NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A SIMULATION TOOL FOR THE DUTIES OF COMPUTER SPECIALIST NON-COMMISSIONED OFFICERS ON A TURKISH AIR FORCE BASE

by

Serhat Camur

September 2009

Thesis Advisor:	Arnold H. Buss
Second Reader:	Susan M. Sanchez

Approved for public release; distribution is unlimited
This thesis was done at the MOVES Institute

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2009	**1. AGENCY USE ONLY *(Leave blank)***

4. TITLE AND SUBTITLE
A Simulation Tool for the Duties of Computer Specialist Non-commissioned Officers on a Turkish Air Force Base

5. FUNDING NUMBERS

6. AUTHOR(S) Serhat Camur

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Staff assignment is one of the major problems in many lines of business. Knowing that the human being is one of the most expensive and demanding resources, efficient personnel employing becomes significant. Simulation techniques can help accomplish effective staff assignments.

The aim of this thesis is to create a simulation tool by using a prototypical model of the computer system specialist non-commissioned officers' jobs on a Turkish Air Force Base, and to identify the effective factors on computer specialist shortage problem. This aim is accomplished by using event graph and discrete event simulation techniques for modeling purposes, and Simkit and Viskit for implementing the created model into simulation code.

After evaluating the simulation results from an experiment involving fifteen input factors, it was concluded that the staff shortage problem can be addressed by using this study after updating the parameters used in the model to reflect the most recent distributions. On the other hand, increasing the number of personnel is not the only solution for addressing the problem. There are some other ways suggested by the study to improve the measure of effectiveness values, such as increasing the number of cars that are assigned to repair personnel, reducing logistic delay times, or increasing the inter-arrival times for computer and network failures. There are different setups or combinations of the factors that are capable of solving the staff shortage problem, and the most cost effective one can be decided after doing a trade-off analysis.

14. SUBJECT TERMS Simulation, Model, Staff Assignment problem, Simkit, Viskit, Design of Experiment, NOLH, and Regression Analysis.

15. NUMBER OF PAGES
97

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**17. SECURITY CLASSIFICATION OF REPORT** Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** UU

THIS PAGE INTENTIONALLY LEFT BLANK

A SIMULATION TOOL FOR THE DUTIES OF COMPUTER SPECIALIST NON-COMMISSIONED OFFICERS ON A TURKISH AIR FORCE BASE

Serhat Camur
Captain, Turkish Air Force
B.S., Turkish Air Force Academy, 1999

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN
MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION (MOVES)**

from the

**NAVAL POSTGRADUATE SCHOOL
September 2009**

Author: Serhat Camur

Approved by: Arnold H. Buss
Thesis Advisor

Susan M. Sanchez
Second Reader

Mathias Kölsch
Chair, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Staff assignment is one of the major problems in many lines of business. Knowing that the human being is one of the most expensive and demanding resources, efficient personnel employing becomes significant. Simulation techniques can help accomplish effective staff assignments.

The aim of this thesis is to create a simulation tool by using a prototypical model of the computer system specialist non-commissioned officers' jobs on a Turkish Air Force Base, and to identify the effective factors on computer specialist shortage problem. This aim is accomplished by using event graph and discrete event simulation techniques for modeling purposes, and Simkit and Viskit for implementing the created model into simulation code.

After evaluating the simulation results from an experiment involving fifteen input factors, it was concluded that the staff shortage problem can be addressed by using this study after updating the parameters used in the model to reflect the most recent distributions. On the other hand, increasing the number of personnel is not the only solution for addressing the problem. There are some other ways suggested by the study to improve the measure of effectiveness values, such as increasing the number of cars that are assigned to repair personnel, reducing logistic delay times, or increasing the inter-arrival times for computer and network failures. There are different setups or combinations of the factors that are capable of solving the staff shortage problem, and the most cost effective one can be decided after doing a trade-off analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

DISCLAIMER

The reader is cautioned that computer program developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the planner.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

API	Application Programmer Interface
DES	Discrete-Event Simulation
DOE	Design of Experiment
FEL	Future Event List
GUI	Graphical User Interface
IFR	Instrument Flight Rules
LANTIRN	Low-Altitude Navigation and Targeting Infrared for Night
MANTIRN	Medium Altitude Navigation and Targeting Infrared for Night
MOE	Measure of Effectiveness
NCO	Non-Commissioned Officer
NOLH	Nearly Orthogonal Latin Hypercube

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank to both Professors Buss and Sanchez for their precious help and guidance through the preparation of the thesis. They made me stay focused when I felt confused and lost.

I also want to thank my beloved wife, Ozlem, for her unwavering support throughout my entire study at the Naval Postgraduate School, and my precious sons, Oguz and Omer. Especially, I beg Oguz's forgiveness because I could not spare enough time to play with him.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

Over the centuries, humans have been obliged to make the best possible decision in the most effective way by running through all the choices. Why has arriving at the best decision been so important? The primary answer to this question is that scarce resources must be used in the most effective way to get the most beneficial results.

In the past, some decision maker's duties were not as hard as they are today because the world was not globalized yet and the options were less varied and sophisticated than current ones. Resource utilization decisions could often be made after considering only a small number of factors, the results of which had been tried before and were well known. A little experience on the job and knowing how to leverage that experience was usually enough for satisfactory results. Moreover, the negative impacts of the wrong decisions were not generally as important as today's and mistakes were easy to recover from and correct.

However, the importance of making accurate decisions has been augmented by the growing relationships among countries and companies worldwide, as well as developing technologies. As a result of these relationships and technologies, research utilization problems have increased and became more complex; more importantly, the impacts of potentially wrong decisions have begun to result in losses that may not be easily recovered from.

This has not only been the case for the civilian sector but also for the armed forces, as well. Commanders have begun to arrive at their critical assessments based not only on their own experiences, but also with the advice of the combat analysts working for them. These analysts often use simulation methods to evaluate the variables among all factors and aspects to inform and support their commanders' decisions.

We could just use mathematical methods (analytical solutions) to evaluate and make decisions about systems if the systems were not as complex as today's are. However, many real world problems today are not simple enough to use analytical methods; therefore, simulations must be used (Law, 2007).

So far, the importance of the simulation has been mentioned. From this point on, the aim of this research will be explained.

Many people have been faced with real or perceived personnel shortages, and have even aired their complaints amongst colleagues. Moreover, they may also have claimed (to colleagues or supervisors) that they have to work harder than normal to cover these personnel shortages. These concerns and claims may be correct in some way. Knowing who allocates personnel to the staff, and how, is important. Do the people in charge of determining the staffing levels use scientific methods?

For many lines of business, the answer is yes. Unfortunately, for the Turkish Air Force, modeling and simulation techniques have not been used at a satisfactory level. One aim of this thesis is to show that these scientific techniques can be used to evaluate the influential factors and obtain more precise personnel assignments.

This thesis will focus on the benefits of determination the number of computer specialist non-commissioned officers (NCOs) needed on a base. It is commonly thought throughout the Turkish Air Force communication battalions that there are not enough NCOs to carry out computer and network maintenance and repair duties due to developing technology and new systems acquisition.

B. RELATED RESEARCH

One staff assignment simulation tool is explained below. It is a thesis research also made by another Turkish Air Force Officer. Basically, it uses discrete-event simulation techniques to determine the required number of personnel to carry out the given tasks.

In this research example, Azimetli (2008) intended to find the number of pilots necessary to meet the increased manpower requirements associated with the introduction of Low-Altitude Navigation and Targeting Infrared for Night (LANTIRN) systems at Turkish Air Force bases (Azimetli, 2008). LANTIRN systems add the capability of flying during night conditions. While this capability dramatically increases the effectiveness of a jet base by providing night flights, it also entails more human resources. Therefore, the Turkish Air Force requested a simulation tool to decide how many more pilots were needed to carry out both day and night missions. This simulation tool is able to determine the number and composition of pilots under certain conditions. Here, composition means the number and proportions of pilots based on flight positions, LANTIRN and MANTIRN categories, and IFR weather categories.

Aside from the category of personnel (computer specialist NCOs rather than pilots), the main difference of this research from the work by Azimetli (2008) is that we use the personnel number as an input factor, where he obtained it as the output of the simulation.

C. RESEARCH QUESTIONS

This thesis research will attempt to answer the following questions:

1. Is the number of the computer specialist NCOs on a Turkish Air Force base large enough to allow them to carry out the duties assigned to them?

2. If, indeed, there is problem in performing the jobs due to a shortage of personnel, then is increasing the number of computer specialist NCOs really the only option? Or, are there any other actions that could be taken to address the problem, such as increasing the number of cars which are used for transportation or decreasing the logistic delay time for carrying out the assigned duties?

D. THE SCOPE OF THE THESIS

This thesis provides an example of the modeling and analysis process for evaluating the computer repair activities at an air force base. The model is based closely based on actual data and operations. The analysis shows how the number of computer specialist personnel required for an air force base can be determined, and it also illustrates how an analyst can assess whether or not increasing the number of personnel is the only solution.

E. METHODOLOGY

The methodologies used in this thesis are listed below:

1. Event graphs and discrete-event simulations will be used to visualize the duties and explain how events are connected to each other.

2. Both Simkit and the visual version of it, Viskit, will be used to implement the event graphs as Java codes.

3. The Nearly Orthogonal Latin Hypercube (NOLH) will be used to build the experimental design.

4. After running the simulation according to the design points created by NOLH, the results of the simulation will be analyzed by the statistical analysis software tool, JMP.

F. BENEFITS OF THIS STUDY

This study aims to show that increasing the number of staff is not the only way to address personnel shortfall problems in many areas, especially in the Turkish Armed Forces. As a matter of fact, it should be considered as the last alternative, after eliminating all other improving factors.

Why is it so important to think twice before increasing the number of personnel? The underlying reason for this is the fact that a human being is the most expensive resource in the world. Moreover, the job tasks require a lot of training, time, and effort before a person is ready for the job.

In Chapter II, the methodologies and modeling tools that are used in this thesis will be explained. A detailed description of the model event graphs and model assumptions appears in Chapter III. Chapter IV contains descriptions of the factor settings and design-of-experiment process used to run the simulation. Analytical results obtained after running the simulation appear in Chapter V, followed by a brief conclusions chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MODELING TOOLS

This chapter gives brief descriptions about the meaning of some key terms, such as system, model and simulation. It also explains the Discrete Event Simulation (DES), event graph methodology, and finally the Viskit software used to create the model.

A. DEFINITIONS OF SYSTEM, MODEL AND SIMULATION

"A system is defined to be a collection of entities, e.g., people or machines, which act and interact together toward the accomplishment of some logical end" (Law, 2007).

It is generally desirable to use the system itself to find a solution for its problems. However, using the system itself may not be cost effective, because it may be difficult to try different combinations of feasible solutions and different combinations of solutions may not be tried easily. Instead, a simulation of the system can be used and, that may give enough insight into solving the problems. First of all, a model of the system should be created to simulate the system. But what does a model mean? A model can be defined as the representation of a system used to study it (Law, 2007).

There are two types of models—physical and mathematical. As Figure 1 shows, either a mathematical model or a physical model can be used to simulate a system. Also, as its name suggests, mathematical models differ from physical ones by using the mathematical representations of the components of the system and the relationships between them.

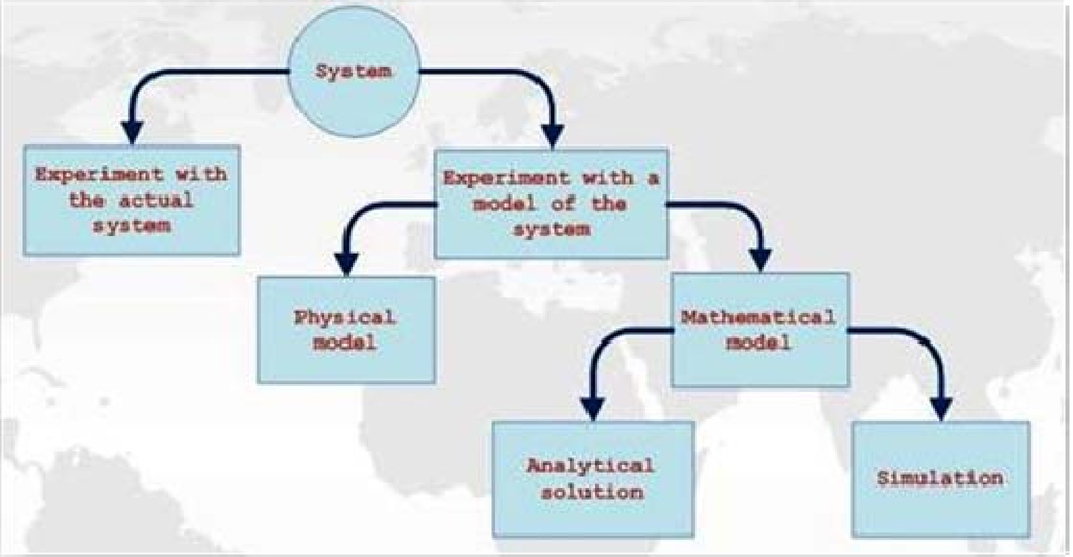

System
Experiment with the actual system
Experiment with a model of the system
Physical model
Mathematical model
Analytical solution
Simulation

Figure 1. Ways to study a system (From Law, 2007)

Because a simulation is just a mathematical representation of the actual system, some key parameters should be defined first. For models of queuing systems, such as manufacturing plants or repair facilities, these parameters can usually be identified as distributional parameters of service times, repair times, and arrival times, as well as the number of servers (such as machines or personnel) or other resources.

Parameter estimates can be obtained by gathering and examining actual system data, or from subject-matter experts. Concluding this examination phase, empirical distributions can be used or suitable distributional models can be derived. After completing all of these efforts, questions such as, "What would happen if it were like this?" can be answered by varying the decision factors. This way, not only can the performance of existing systems be increased, but the systems that are still in the planning phase can also be designed to operate more effectively. Simulation technology holds tremendous promise for reducing costs, improving quality, and shortening the time-to-market for manufactured goods (McLean & Leong, 2001), and similar benefits are possible for improving repair facility operations.

A simulation can be defined as deterministic or stochastic according to its inclusion of randomness. Deterministic simulations have no randomness in them, thus if the inputs are held constant, the resulting outputs are always the same. They do not change from one run to another. In this thesis, a stochastic simulation will be used. This means that each combination of inputs must be replicated (that is, run several times), and the output will be analyzed using statistical techniques.

As mentioned above, simulation is a technique to create realistic models of the systems to assist in a decision-making process. Once an appropriate model has been constructed, running the simulation on a computer and then using statistical tools to evaluate the results can lead to useful insights. By using a well-designed experiment to specify an appropriate set of simulation runs, the analyst can gain these insights much more quickly and effectively than by using a trial-and-error approach.

B. DISCRETE-EVENT SIMULATION

Law states that the discrete-event simulation paradigm models a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time (Law, 2007).

The state variable changes are caused by the events, which occur at discrete times (Schriber & Brunner, 2005). For example, the number of personnel in a computer repair center is a state variable that increases after a computer is repaired. Likewise, when a specialist grabs a malfunctioning computer for repair, this decreases the number of available personnel by one.

1. Components of the Discrete-event Simulation

The following components are used in most real-world problems, independent of their kinds (Law, 2007).

a. System State

As mentioned earlier, some variables change at discrete times during the simulation. So, the system state represents the complete set of all state variables' conditions at a given time.

b. Events and Parameters

An event is defined by Law as an “instantaneous occurrence that may change the state of the system (Law 2007).” As mentioned earlier, a “repair event” may increase the numbers of computers and computer repair personnel available.

Parameters are constants and do not change when the simulation advances. In other words, they do not have states. For instance, the mean time that passes to repair a computer is a parameter and it does not change, like state variables, when events occur. Note that in a stochastic simulation, even though the mean repair time parameter is constant, the actual repair time is typically a randomly generated value from a distribution with this mean. Normal, uniform, and exponential distributions are often used for modeling purposes.

c. Event Lists

State variables change when an event occurs. However, changing the state variables is not the only job of an event—they also schedule other events in the simulation. Therefore, there is a need to keep track of the upcoming events to be executed when it is their turn. The event list does this job, and shows the sequence of events to be executed. For instance, in the computer repair center example, an “end repair” event may add a “start repair” event to the list if the necessary conditions are satisfied—that is, if there are more computers to be fixed.

C. EVENT GRAPHS

“Event graphs are a way of graphically representing discrete-event simulations” (Schruben, 1983). These graphs are also known as simulation graphs (Schruben & Yücesan, 1988). This event graph methodology is used in many discrete-event simulations.

There are two main reason for using event graphs for representing problems. These are simplicity and, despite their simplicity, their power to represent even complex problems (Buss, 1996).

An event graph consists of nodes and edges. Events are represented as the nodes (here, shown as circles) in the event graph. As stated before, each event may consist of a state transition. Each edge corresponds to scheduling another event. Also, each edge connector may or may not have a boolean variable that controls the execution of the other events.

As stated before, the advantage of event graphs is their simplicity. After understanding the basic concepts, it is not difficult to model more complex systems. For instance, Figure 2 shows the basic structure of event graphs. Events and state transitions are showed as circles. Arrows are used to schedule other events. The notation "t" represents the delay time between two events. Finally, a wavy line shows the condition that should be obtained to schedule another event. Figure 2 can be interpreted as follows: "the occurrence of Event A causes Event B to be scheduled after a time delay of *t*, providing that the condition (i) is true (after the state transition for event A has been made) (Buss, 1996)." If there is no condition and no delay, Event B is always executed without any delay when Event A is executed.

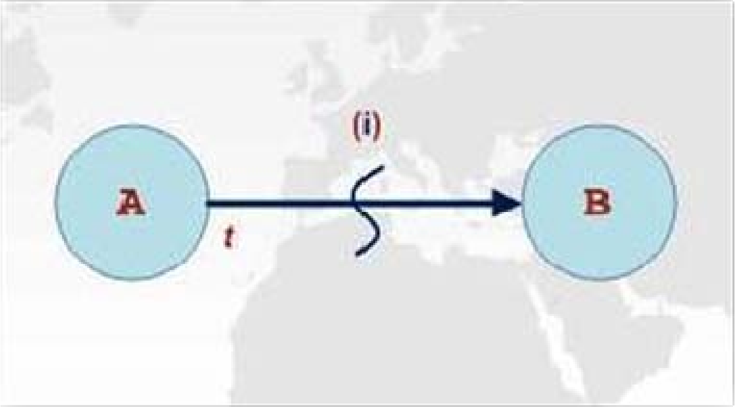


Figure 2. Fundamental event graph construct (From Schruben, 1983)

Figure 2 represents the basic construct for event graphs but, in reality, this may not be enough. Some enhancements should be introduced to deal with more complex problems. Two of the important enhancements are "passing attributes to events on scheduling edges" and "event-cancelling edges" (Buss, 1996).

1. Passing Attributes on Edges

With this feature, it is possible to pass attributes from one node to another. For example, an attribute can be a failure entity and may need to be passed for the calculation of time in the system. The only difference between Figure 1 and Figure 2 is the added attribute *k*.

The interpretation of Figure 3 is, "when event A occurs, A's state transitions are made and expression *k* and condition (i) evaluated. If condition (i) is satisfied, then event B is scheduled to occur after a delay of *t* time units with parameter *j* set equal to the computed value of *k* (Buss, 1996)."

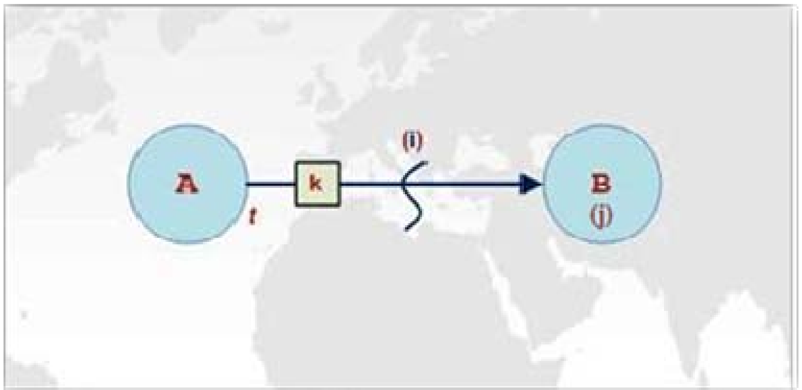


Figure 3. Passing attributes on edges (From Schruben, 1995)

2. Cancelling Edges

This enhancement deals with situations such as a scheduled event that needs to be removed from the event list (Schruben, 1983). For this study, the most significant example of this requirement would be customers waiting in a queue for some time but then leaving without getting service in their scheduled time. This feature is represented with the dashed arrows in the event graphs.

D. VISKIT

The previous sections explained the modeling of a problem with event graphs and the underlying techniques to fulfill this job.

Now it is time to implement these previously created event graphs into simulation code and obtaining the statistical results after running that code. This can be done using the component-based simulation Java package called “Simkit” written by Prof. Arnold Buss (Buss, 2002).

In this first version of Simkit, the simulation modeler had to interact with Simkit at the Application Programmer Interface (API) level (Buss, 2002). However, with the new version, a graphical interface has been provided for creating the simulation easily and more intuitively. This version is called “Viskit.”

“Viskit is a graphical front end for creating, editing, and composing DES simulation models using event graphs and the Listener Event Graph Objects (LEGO) framework” (Buss, 2007).

1. Event Graph Editor

The event graph editor is used to draw the components of the model prepared as event graphs. Basically, the same types of shapes and arrows are used in Viskit as those just described for representing event graphs. This makes the implementation phase easier.

The event graph editor has four main sections: a palette to draw the event graph by using the nodes and arrows provided as a separate section above, a section for defining the state variables, a section for defining parameters, and, finally, a panel named “Code Block” that adds more functionality and flexibility into event graphs. A screenshot of the event graph editor is provided in Figure 4.

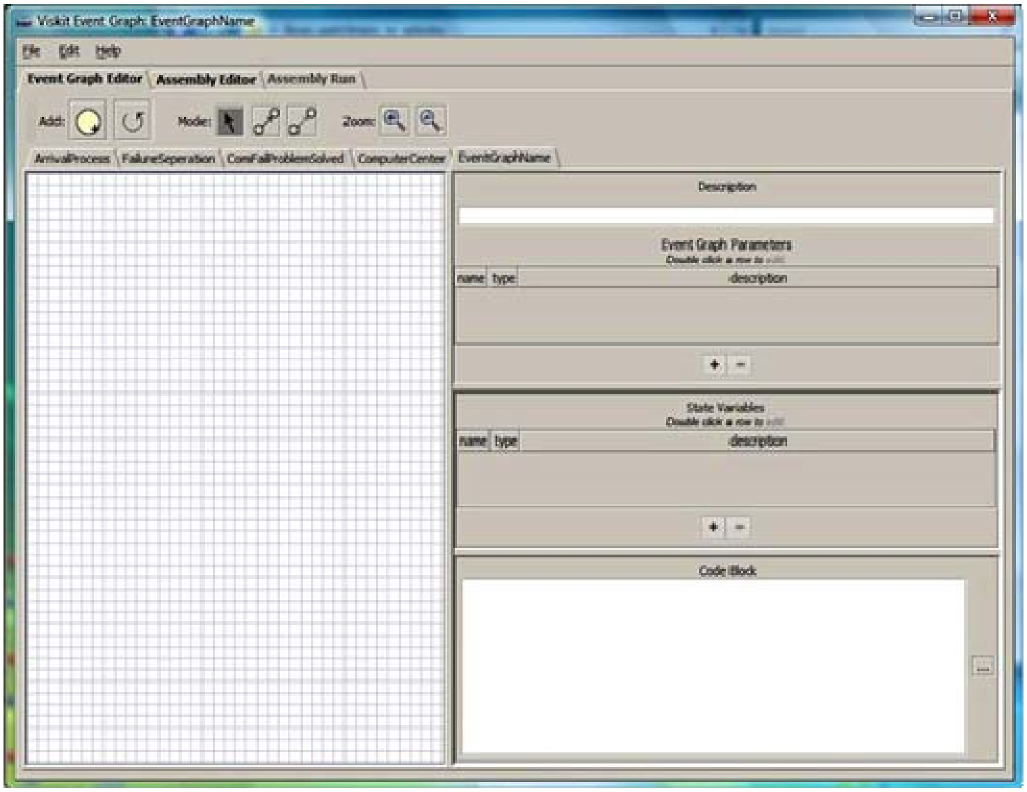


Figure 4. A screenshot from the Viskit event graph editor

a. Node (Event) Inspector

Figure 5 shows the node inspector used to input the data associated with the node (Buss, 2007). Simply, these are the modifications that can be done for a node:

(1) Changing the name of the node,

(2) Adding a description,

(3) Implementing event arguments, local variables, and state transitions,

(4) Implementing a code that is needed to run when this node is executed.

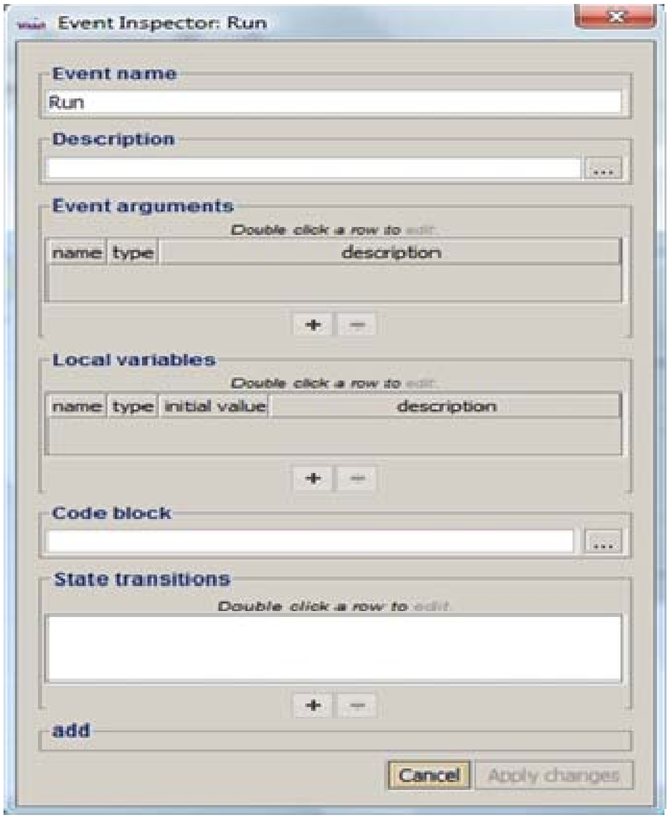


Figure 5. Viskit node inspector

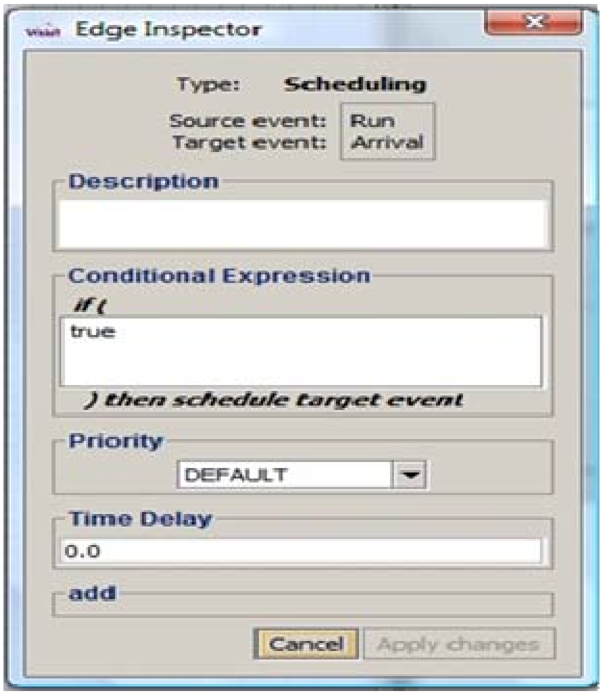
b. Edge Inspector

The edge inspector shown in Figure 6 is used to input information about the edges:

(1) Adding a description,

(2) Defining a conditional expression,

(3) Providing a time delay that can be either a fixed value or a random variable. Random variable is defined as a parameter and an instance of it got for the time delay.



Edge Inspector

Type: Scheduling

Source event: Run

Target event: Arrival

Description

Conditional Expression

if (

true

) then schedule target event

Priority: DEFAULT

Time Delay: 0.0

add

Figure 6. Viskit edge inspector

2. Assembly Editor

After creating the event graphs by using the event graph editor, it is now time to link those components to each other, and, finally to provide a way to gather any statistics or state values of interest, such as mean values and counts. This is done on the Assembly Editor, an example of which is shown in Appendix A.

As previously mentioned, the model is divided into small reusable pieces and drawn by using the event graph editor. Therefore, components should be connected to each other to make an event that will trigger the connected element in another component. To achieve this functionality, Viskit provides mechanisms called "listeners" and "adapters."

The first mechanism is the “listener.” To use the “listener pattern,” there should be identical (in both name and signature) event nodes in both components (Buss, 2009). When an event occurs within a source, that event triggers the same event in the listener component.

The second mechanism is the “adapter pattern.” In this mechanism, there is no need for the events in the source and listener components to be the same. Instead, when an “adapter” is used, the source and listener events should be entered explicitly by the user.

After creating the model by using the assembly editor, some statistics should be added. This is done by choosing the appropriate statistical function from the “Property Change Listener” section of the assembly editor.

The simplest property change listener is “Simple Property Dumper.” This listener lists the state variable changes in the components that it is connected to by a connector (the pitchfork-like button on the top section) and writes them on the screen.

There are also some other statistics features of Viskit. These are “CollectionSizeTimeVaryingStats,” “SimpleStatsTally,” and “SimpleStatsTimeVarying.” These are used for getting count or mean statistics from the containers (“LinkedLists,” etc.) or state variables.

Next chapter will present the event graphs of the system and their descriptions.

THIS PAGE INTENTIONALLY LEFT BLANK

III. MODEL EVENT GRAPHS AND DESCRIPTIONS

Now that the basics of event-graph modeling have been explained, details about the computer repair model can be provided. We begin with a very brief description of the system, followed by explanations of assumption that were accepted before the modeling process began. With the assumptions in mind, event graphs and their roles in the model, and finally the assembly created using the Viskit simulation program, will be explained.

A. ASSUMPTIONS

1. In this thesis, two types of personnel are created: experienced and inexperienced. However, personnel expertise may be more complex in real-life conditions.

2. When personnel must travel to get to failure locations, they are assigned cars to drive. Cars are considered to be up and ready all the time in this model; that is, car failures are not taken into consideration.

3. Personnel are assigned to the failures on a one-for-one failure basis. Some tasks may require more than one person in real life.

4. There is no limitation for answering the phone calls. If there are available personnel, a call is answered. In reality, this may be limited by the number of the telephones in the center.

5. As mentioned earlier, cars are the only type of vehicle used for transportation purposes. Although many bases provide a ring system that includes buses that tour the base on a regular schedule, this capability is not included in our model.

6. The only resource consuming events in this model are associated with personnel attending training courses or taking vacation leave. Other resource-consuming events are not included.

7. Experienced personnel are assigned to repair jobs whenever possible. Inexperienced personnel are used only if all the experienced personnel are either away from the base or busy assisting other customers.

B. MODEL EVENT GRAPHS

All parts of the model used to run the simulation and obtain results are explained in detail in this section.

The first four classes are created in Netbeans using Simkit and integrated into Viskit as a Jar file. This feature of Viskit is very useful for the simulation parts that cannot be generated or are hard to generate in Viskit.

1. Server Entity

In a base, computer and network failure maintenance and repair jobs are performed by computer system specialists. These specialists are NCOs in the Turkish Air Force. These NCOs are generally chosen from among the graduates of business high schools. After being chosen, they get a two-year education, which includes both military and computer training. After this two-year period, they graduate and are assigned to bases all over Turkey. Thereafter, they are sent to various follow-on courses to adapt to new technologies and to increase their excellence.

Therefore, their experience levels may change based on their background and their own efforts. The experience level affects service and inspection times when they are assigned to a maintenance job or a failure.

Hence, to deal with this experience issue, two kinds of personnel are used in the simulation—experienced and inexperienced.

This differentiation is made by using Java enums. Enums types are useful for creating a fixed set of constants, such as compass directions (NORTH, SOUTH, EAST, and WEST) (The Java Tutorials n.d.).

In this class, two entities, EXPERIENCED and INEXPERIENCED, are created. Also, these types have the features of the Simkit Entity class.

2. Server Comparator

As its name suggests, this class is used to compare server entities. Experienced personnel were assigned to the requesters in first place if there was availability.

3. Times

This class is created to handle all the times used in the simulation. Most of the components (graphs or classes) need to use similar time values through the course of the simulation. That is why a separate time class is created. If a component requires a time value, it may get it by creating an instance of the times class and getting the appropriate time by using the field variable obtained. A brief description of each time class is given below.

a. Service Times

This is a Simkit RandomVariate variable, which is different from the regular variables, such as double or integer. These variables get their values based on a distribution like “Exponential” or “Gamma.” Simply, to handle the required parameters for these distributions, time values should be obtained for a period of time and the distribution type and its value (e.g., mean) should be obtained.

The service times are different for different experience types.

b. Inspection Times

InspectionTimes account for the time that an individual uses to determine the type of failure and the parts needed to repair it. These times are also RandomVariate variables, and are obtained in a similar way as explained above in ServiceTimes.

There is also a difference between experienced and inexperienced personnel in terms of times spent on inspection.

c. Phone and Remote Access Time

When a user experiences a problem with his computer, the first thing he does is to call the computer center and consult a specialist about the problem by phone.

This may be helpful, and end up with the computer repaired either by phone or by remotely accessing the computer. Specialists ask questions to understand the problem and give directions or, if there is not a problem with the network connection, may connect to the computer remotely.

d. Times to Get to Site and Come Back to Center

These times are implemented to simulate the transportation delay experienced in getting to a failure location and coming back to computer center after finishing the job at the failure site.

e. Small Failure Repair Time

Sometimes users call the computer center with a failure due to a lack of basic computer and networking knowledge that they may be able to repair themselves. At these times, repairing the problem does not take much time. Therefore, another ServiceTimes value is implemented to deal with small failures. This time is same for all specialists.

f. Logistic Delay

This time accounts for logistics delays arising from shortages of computer or network parts for repair. At times like these, the supply section acquires the needed parts from the market. Therefore, this causes some logistic delays.

g. Course Times

As mentioned earlier, from time to time, personnel attend various training and educational courses to increase their expertise and also adapt to the new technologies. The durations of these courses depend on the type of the course.

4. Personnel Server Source

Like the “times” classes, this class or event graph deals with personnel issues such as assigning personnel to the requesting source, receiving personnel back from those sources after they are finished, and making the state transitions when these events occur. Figure 7 may help understanding this mechanism.

Personnel are held in a container. When some source requests personnel (Server Source listens to the other classes by a Simkit listener), this class looks at the personnel container and available cars, since a car is needed to get to the failure locations far from the computer center. If both are available, then the most experienced individual is assigned (other classes listen to this class). If either personnel or car is not available, this class adds the requester source into another container and personnel are assigned when they become available. During this time interval, the requester waits or listens for the personnel from the source server class.

However, all requests do not necessarily require waiting for a car. For instance, when the graph for the “personnel leaves” requests a source, it does not need to check the car availability because car availability is just needed for the transportation to the failure locations. If there are personnel in the container, then that entity can be assigned. This is done by using the “property setting and getting” mechanism that is provided within the Simkit entities. Basically, requesting sources carry their car requirement information by their assigned property.

There are some parameters within this class, as well. These are “total personnel,” “total number of cars,” “percentage of experience personnel,” and “rest time” for personnel returned from a job. “Total personnel” shows the number of personnel, given as a parameter, and the “experience percentage” is used for calculating the number of experienced personnel. When personnel return from a job (Server Source class listens to the other classes by a listener pattern), they rest for some amount of time. Finally, the “total car” parameter shows the number of cars earmarked for the use of the computer center.

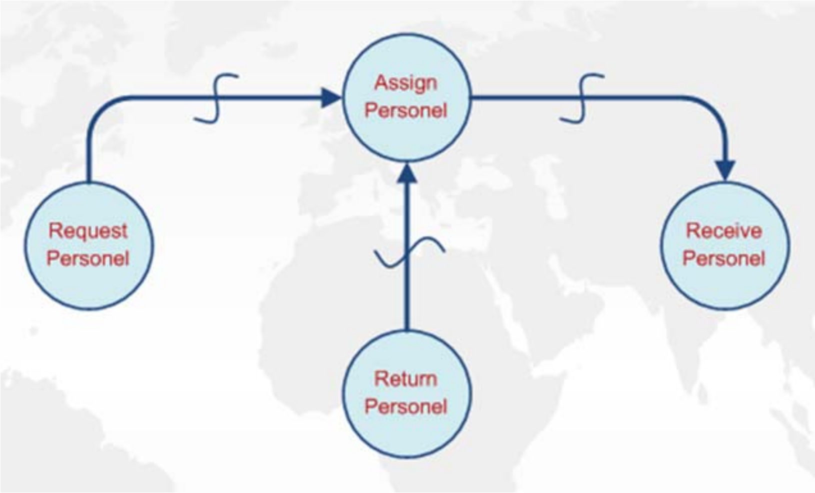


Figure 7. Personnel server

5. Arrival

An example of the arrival class is shown in Figure 8. It has a parameter called "inter-arrival time." This "inter-arrival time" changes depending on the usage of this component. That is, times will be different for computer and network failures.

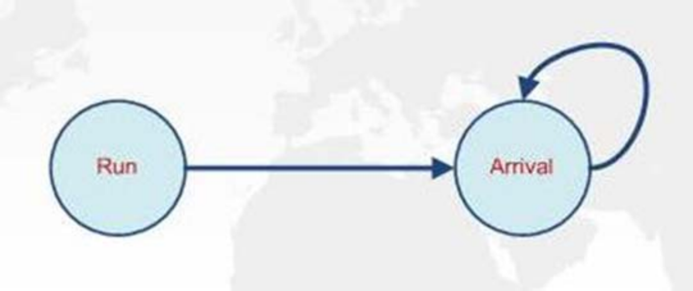


Figure 8. Arrivals

6. Entity Arrival

This event graph listens to the arrival event graph, and when an arrival event gets executed in the event list, it schedules an “EntityArrival” event. A new entity is created when an “EntityArrival” event occurs, and this is passed to the listening graphs as an argument.

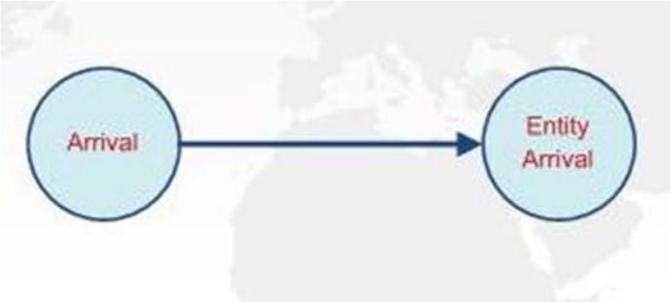


Figure 9. Entity arrivals

7. Computer Failure Arrival

This class is connected to the “EntityArrival” class by a Simkit adapter. Every entity arrival event in the “EntityArrival” graph schedules a computer arrival event in this class. Entities are created here when an arrival computer event is executed. These failure entity arrivals are kept in a container until personnel are obtained from the personnel server source class. That is why a “failure arrival” schedules a “request personnel” event.

As is shown in Figure 10, after receiving personnel by the receive personnel event, the decision of whether the problem will be solved over the phone or by remote access is made by drawing a random uniform number (0, 1) and comparing it to the given probability, that is, only some portion of the problems may be solved by remote access, and the systems can’t be solved should be repaired at the failure location or at the computer center.

In both cases, whether the problem has been solved or not, a time delay that is generated by using the times class explained earlier is added to the total time passed since the entity was created. As a reminder, entity creation times are stamped on them when they are created.

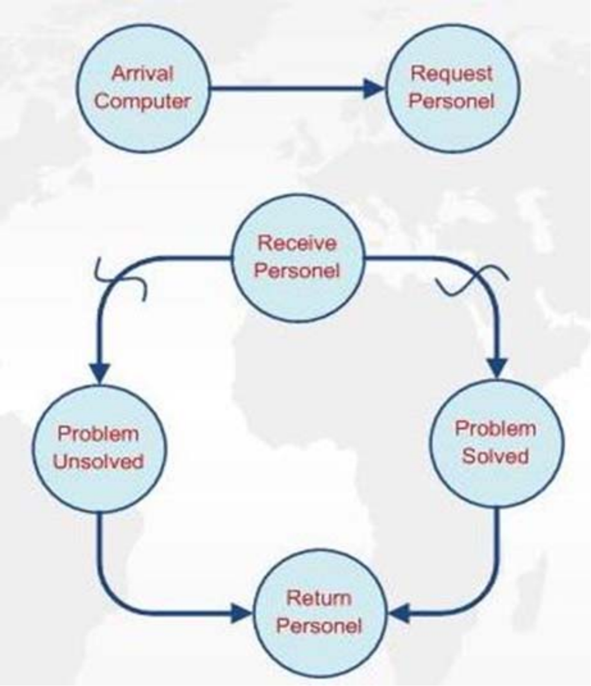


Figure 10. Computer failure arrivals

8. Problem Unsolved Remotely

As mentioned earlier, sometimes a computer specialist can resolve a problem merely by talking on the phone with the user or remotely accessing the malfunctioning computer. This class listens to the situation in which the problems could not be resolved.

After receiving the igniting event and not resolving the problem, the computer center will either request the user to bring the computer to the center for repair, or send a computer specialist to the failure location. Generally, this decision is made by the technician who tried to repair the issue remotely, based on the impression that he developed in the unsuccessful repair attempt.

As shown in Figure 11, personnel are requested from the personnel server, and when the technician is obtained, a "RepairAtSite" event is scheduled by a transportation delay.

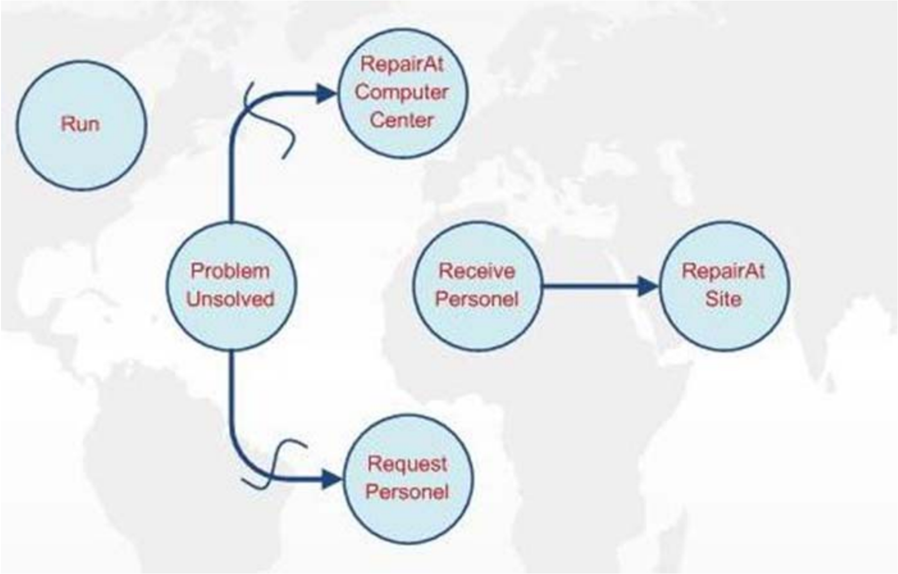


Figure 11. Remote access

9. Begin Repair at Failure Location

As is shown in Figure 12, when a repair at location event is heard, an inspection event begins instantly and ends after a time is obtained from the times class.

After inspection, the size of the problem is decided by comparing a random uniform number with the probability of a big failure.

If the failure is small, it is repaired at the failure location by the technician. If it is big, the computer is taken to the computer center and added to the queue for repair.

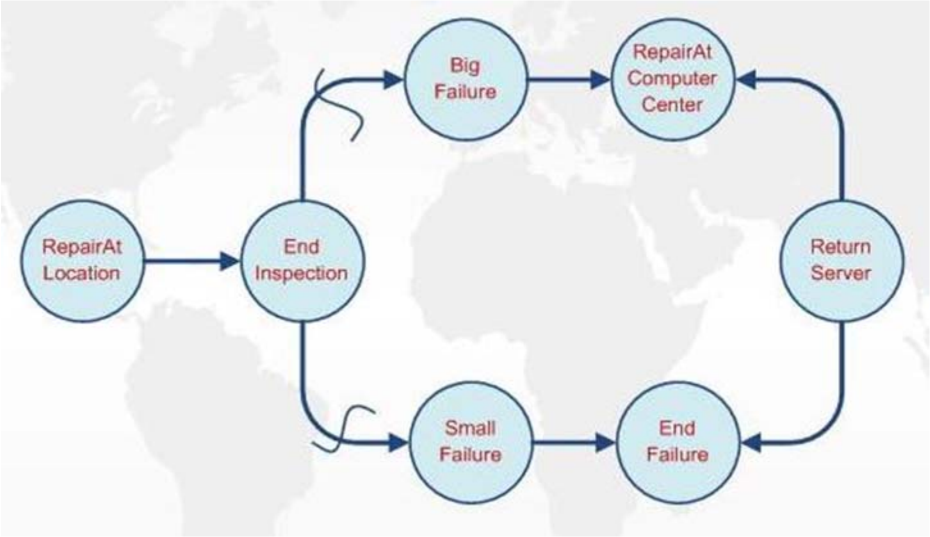


Figure 12. Repair at site

10. Decision for Adding Logistic Delay

Again, a personnel request is first made from the personnel server. After acquiring a technician via the “Receive Server” event shown in Figure 13, the computer is inspected. After this inspection period, the broken parts are identified and, if there are enough parts for repair, there is no need for logistic delay. Otherwise, a logistic delay time will occur. The need for a logistic delay determination is made by generating a random number and comparing it with the probability provided.

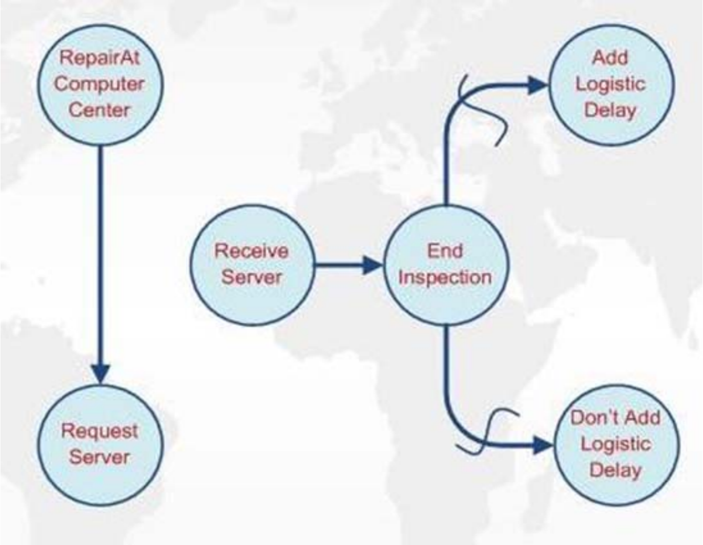


Figure 13. Decision to add logistic delay

11. Repairing with Logistic Delay

After deciding that some parts are needed to repair the computer, the assigned personnel should be returned to the server immediately. Because there will be a logistic delay, not returning the personnel will affect personnel utilization. Thereafter, another personnel request without a car should be done for repairing the failure at the computer center (Figure 14). In earlier graphs, car need decisions were done while requesting a personnel from the server source. Thus, when the personnel server gets this source, it assigns personnel without considering the car status.

When a technician is assigned, the repair is ended after the period of time obtained from the times class.

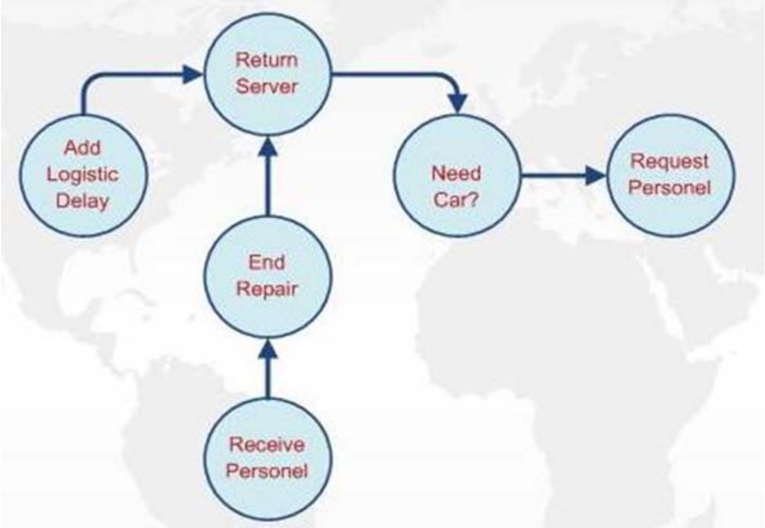


Figure 14. Repair with logistic delay

12. Repairing without Logistic Delay

The only difference between this and the previous event graphs is that a logistic delay is not added here (Figure 15). Otherwise, all is the same as before.

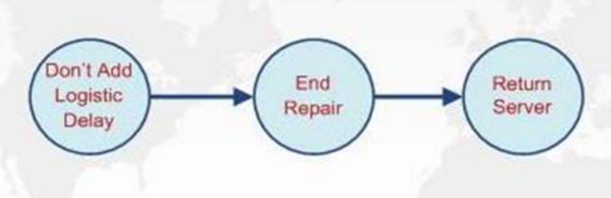


Figure 15. Repair without logistic delay

13. Network Failure Arrival

The logic for network failures is similar to the computer failures explained previously.

After getting a failure event, personnel are requested from the personnel server. If there are available personnel that can be assigned to the source, then the server assigns one. Thereafter, “start repair,” “end repair,” and “return server” events are executed.

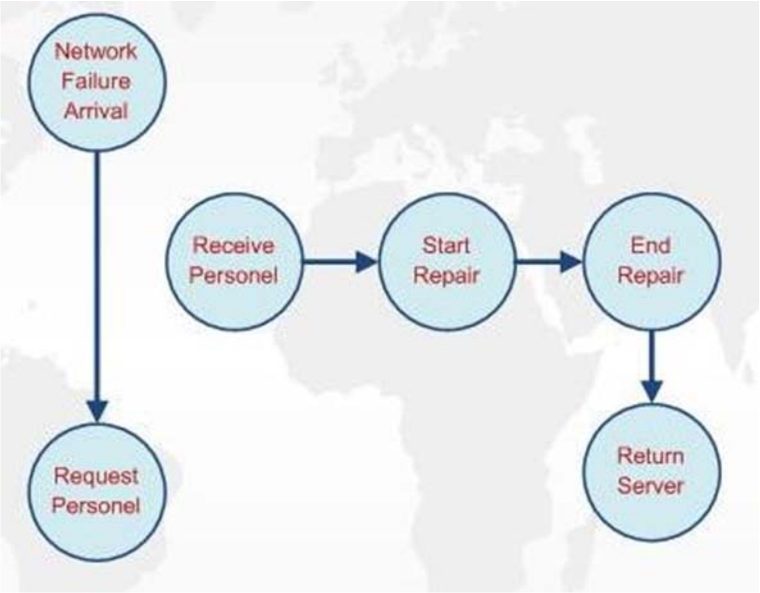


Figure 16. Network failure arrivals

14. Leave Graph

When considering “leaves” for vacation, the events until getting the personnel from the server are similar to the previous event graphs.

After obtaining the personnel, it should be decided whether the leave request will be approved or not. That is, if certain conditions are not met, a leave request may not be approved and, in that case, the personnel should return to the server immediately.

What are these conditions? One of them is that the total personnel on leave should not exceed a given threshold. If this first condition is successfully met, then the appropriate personnel properties are checked. The properties are winter, summer, and daily leave tracks. They are set to zero when the personnel are created at the beginning of the simulation with regard to the parameters defined. Typically, personnel have 10 days for winter, 20 days for summer, and a changing rate for daily leaves.

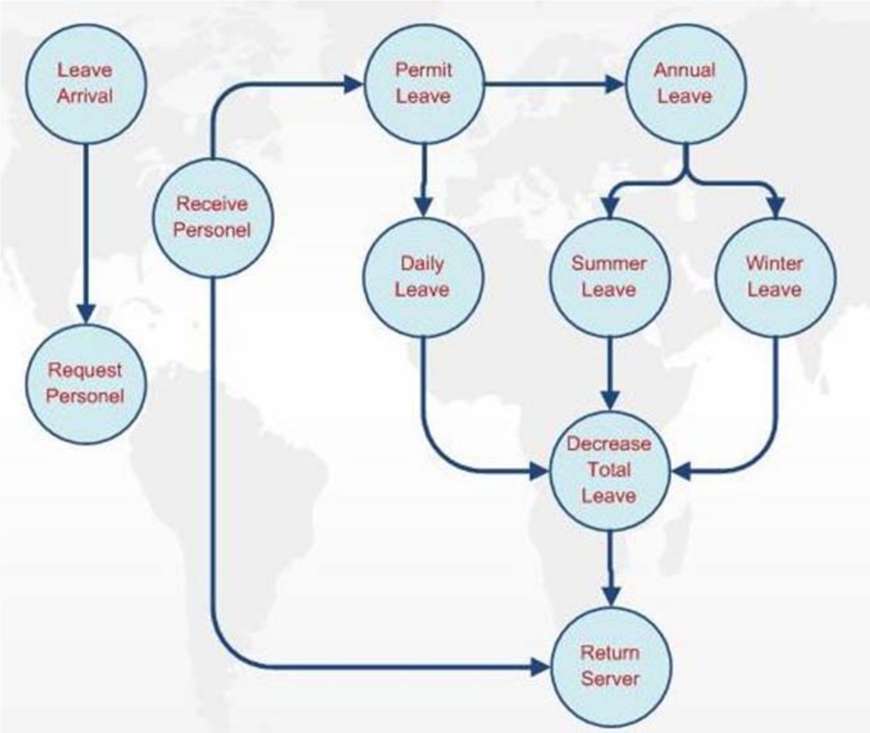


Figure 17. Leave graph

15. Course Graph

As in the “leave graph,” there are also some conditions in this graph that should be met to send the personnel to a training course.

First of all, a total annual number of courses are defined at the beginning of the simulation. That threshold should not be exceeded. Next, the number of personnel taking some kind of course should not exceed a given threshold.

If these conditions are met, then the personnel can attend the course and return to the server after a length of time obtained from the "times" class.

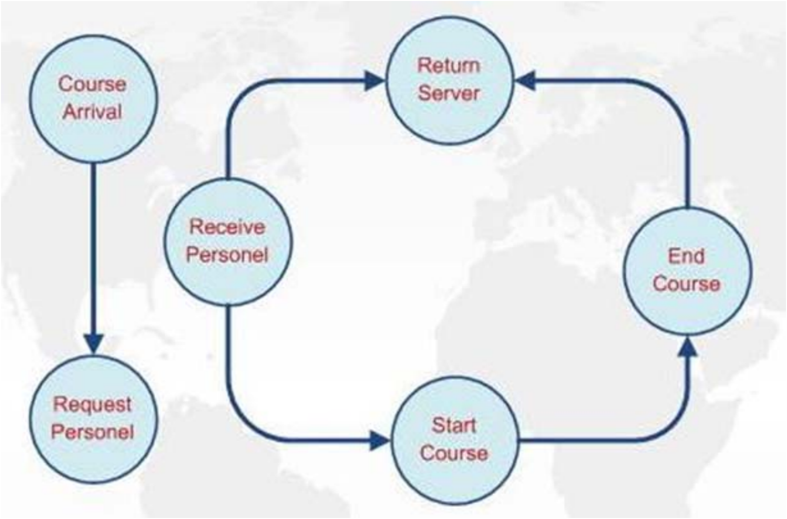


Figure 18. Course graph

C. MODEL ASSEMBLY AND RUN

As mentioned previously, after creating event graphs (components), these components should be connected to each other by the listener or adapter patterns. Following creation, the assembly should be run after inputting the appropriate parameters.

Part of the assembly used to run the simulation is shown in Appendix A. In the assembly, there are four arrivals: "computer," "network," "leave" and "course." Only one of them will be explained in detail, since the others have similar features. Computer failure arrivals are taken into consideration as the example. Entities are created in "computer entity" node. This node listens the "computer arrival" node by using a listener pattern that connects those two nodes. "Computer graph" is connected to "computer entity" node with an adapter pattern to be aware of created entities and doing the following jobs. "Computer graph" is also connected to "server source" with three adapters that are used for personnel request, receiving the assigned personnel, and returning the personnel back to personnel server source, respectively.

After describing the event graphs and the assembly, the next chapter will describe how to design the experiment in order to collect data for output analysis.

IV. DESIGN OF EXPERIMENTS

This next section defines fifteen input factors of interest for the computer center. These factors are anticipated to influence the measures of effectiveness described in Section B. In Section C, the benefits of using an efficient experimental design are explained. In the final sections, we describe the design (i.e., systematic combinations of input factor settings) used to run the simulation experiment, and discuss the need for replicating the design points.

A. INPUT FACTORS

Fifteen input factors, which will next be described in detail, are varied systematically to obtain insights into the computer repair facility staffing process. The minimum and maximum values of the parameters are based on the information obtained from a Turkish Air Force Jet Base Computer Center for some factors and on the experience of the author for some others.

1. Experienced and Inexperienced Personnel's Service Times

These two input factors are used to measure the effects of experienced and inexperienced personnel on the Measure of Effectiveness (MOE) values. As it is known, the triangular distribution has minimum, maximum and a mode value. These times were considered as the minimum and maximum. In other words, these values stay constant whereas the mode value changes between these minimum and maximum values based on the design of experiment.

a. Experienced Personnel Service Time

(1) Minimum: 1 hour

(2) Maximum: 2 hours

b. Inexperienced Personnel Service Time

(1) Minimum: 1.5 hours

(2) Maximum: 3 hours

2. Experienced and Inexperienced Personnel's Inspection Times

These two factors are similar to service times but used for the inspection period.

a. Experienced Personnel Inspection Time

(1) Minimum: 30 minutes

(2) Maximum: 45 minutes

b. Inexperienced Personnel Inspection Time

(1) Minimum: 48 minutes

(2) Maximum: 84 minutes

3. Small Failure Repair Time

When a computer specialist goes to a failure location, he decides whether a problem is a big or a small one. If it is decided that the problem is small and can be repaired at the site, then it is repaired with the delay that is generated from this variable.

There is no difference between experienced and inexperienced personnel for this time factor. These times were considered as the minimum and maximum values of the triangular distribution. In other words, these values stay constant whereas the mode value changes between these minimum and maximum values based on the design of experiment.

a. Minimum: 18 minutes

b. Maximum: 60 minutes

4. Logistic Delay Time Mean and Delay Probability

This factor is used when the repair cannot be made because parts are lacking. This is decided by a probability defined in the simulation. If a logistic delay is identified, then the repair process is delayed for a period of time. These time values were considered as the minimum and maximum values of the triangular distribution that address a possible logistic delay time. Again, these values stay constant whereas the mode value changes between these minimum and maximum values based on the design of experiment.

a. ***Logistic Delay Time***

(1) Minimum: 4 days

(2) Maximum: 16 days

b. ***Logistic Delay Probability***

(1) Minimum: 0.3

(2) Maximum: 0.6

5. The Number of Total Personnel and Experienced Personnel Percentage

The total number of personnel and the percentage of those who are experienced are important factors in the simulation. These two factors are expected to affect time and queue waiting calculations notably.

a. ***Total Personnel***

(1) Minimum: 9 personnel

(2) Maximum: 15 personnel

b. ***Experienced Percentage***

(1) Minimum: 0.3

(2) Maximum: 0.8

6. Total Number of Cars

Cars are used for transporting the computer specialists to the failure locations. Most of the delays may occur due to a lack of cars, even if there are personnel on hand to assign a job.

a. Minimum: 1 car

b. Maximum: 3 cars

7. The Probability of Solving the Problem by Phone or Remote Access

Internet service providers tend to connect customers' computers to solve the Internet connection issues before taking another action. Similar to that, a user in one of the sites on a Turkish Air Force base calls the computer center to consult about the problem when he gets a failure. This is the first step of the repair process, and some problems can be solved by talking on the phone or remotely accessing the computer or network.

a. Minimum: 0.3

b. Maximum: 0.6

8. Computer and Network Failure Arrivals

These two failure arrivals are also chosen to be an input factor. The first one represents a computer failure and the second one represents network failure arrivals. These values show the inter-arrival times between failures. Therefore, larger values are better. The distributions of both types of inter-arrival times are modeled as exponential distribution.

a. Computer Failure Arrivals (Mean)

(1) Minimum: 3 hours

(2) Maximum: 7 hours

b. Network Failure Arrivals (Mean)

(1) Minimum: 8 hours

(2) Maximum: 15 hours

9. Minimum Required Personnel on Base

This parameter is used to decide whether to give daily leave permission to personnel. This is an important parameter because in real-time conditions, commanders may not permit personnel to take leave if the number of personnel is under a given threshold. Note that both these numbers are far less than the minimum staff for the center.

a. Minimum: 2 personnel

b. Maximum: 5 personnel

10. Total Daily Leave

This is regular leave given to personnel. It is highly dependent on the job intensity at a site. Thus, it may change from site to site.

- ***a.*** Minimum: 9 days per person per year
- ***b.*** Maximum: 15 days per person per year

B. PERFORMANCE MEASURES

1. Mean Delay Time in System

When an entity is created during the simulation, its creation time is stamped and carried over wherever it goes. After an entity gets repaired in some way during the simulation, the time passed until that time is calculated. This time value shows the delay in the system. This value should be short for a system to be effective and return the down system to the user in short time.

2. Mean Number in the Queue

This MOE shows the mean of failures waiting in the queues across the simulation. This is also an important measure since short queue lengths are also desirable.

C. NEARLY ORTHAGONAL LATIN HYPERCUBE (NOLH)

A successful analysis requires an effective experimental design to get the maximum benefit from the simulation runs (Sanchez, 2009). Within the many available experimental design options, the factorial design may be the most familiar one.

A 2^k design is a factorial design, where 2 shows the number of levels for each of the *k* input factors during the experiment, and the number of design points is $N = 2^k$. Generally, levels are represented as on or off, low and high (Sanchez, 2008). Figure 19 shows an example of 2^2 factorial design. Here the number of the design points is calculated as $N = 2^2$. Therefore, for this design, there are four design points.

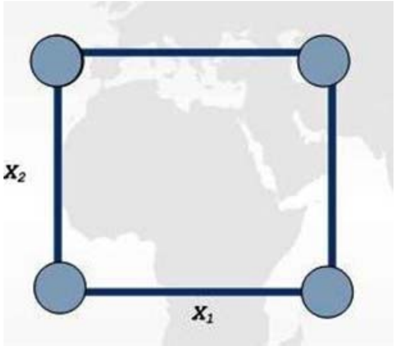
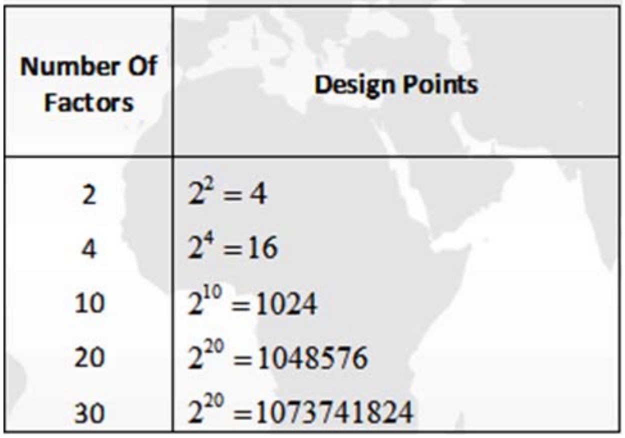


Figure 19. A representative 2^2 factorial design

There are many useful sides of factorial design, like being easy to build, being orthogonal, and allowing the researchers to inspect and determine the main effects and the interactions between them.

However, it may be inefficient or impossible for some cases, as is shown in Table 1. The number of design points grows exponentially with the increasing factors. Moreover, it does not represent or explain what happens at interior points; it only deals with the corners as it is shown in Figure 19. To increase the coverage, the number of levels of the factors can be increased from 2. But this causes the number of design points to grow even more dramatically.

Table 1. Required design point numbers for various factors in 2^2 factorial design



Number Of Factors	Design Points
2	$2^2 = 4$
4	$2^4 = 16$
10	$2^{10} = 1024$
20	$2^{20} = 1048576$
30	$2^{20} = 1073741824$

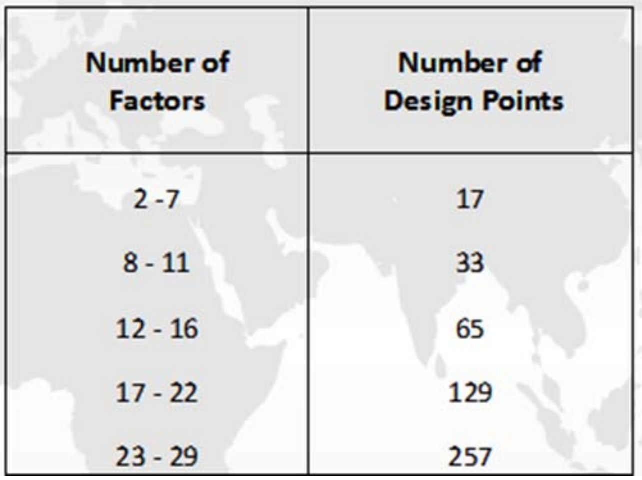
Therefore, using the NOLH design developed by Cioppa and Lucas (2007) may decrease the negative effects of other designs while preserving the positive features, as explained below (Cioppa & Lucas, 2007).

Some of the advantages of NOLH design are:

- Efficiency,
- Space-filling feature,
- Design and analysis flexibility.

Efficiency means that they require far fewer design points than many other experimental designs, as can be seen by comparing the numbers shown in Table 2 (for the NOLH designs) to the numbers in Table 1 (for the 2^k factorial designs).

Table 2. Required design point numbers for various factors in NOLH design



Number of Factors	Number of Design Points
2 -7	17
8 - 11	33
12 - 16	65
17 - 22	129
23 - 29	257

D. DESIGN POINTS

Since NOLH design provides a very efficient experimental design, it was used to design the experiment. For the 15 input factors, an NOLH requires 65 design points. But for capturing more data and getting more precise results, this number of design points can be essentially doubled by combining two NOLH designs. The second design is constructed from the base design by changing columns of factors in the NOLH spreadsheet, and the designs are stacked. As a result, 129 design points (the middle design point is the same for both designs so one of them is removed) are used as inputs the simulation.

To illustrate the space-filling property of the NOLH design, a scatter-plot matrix showing the pairwise projections of some of the input factors is shown in Figure 20. As seen from the figure, this NOLH design is notably good at filling the space and representing many combinations of input factor settings—particularly for the continuous factors. The input factor minimum and maximum values and the design points obtained from the NOLH spreadsheet (Sanchez, 2005) is presented in Appendix B.

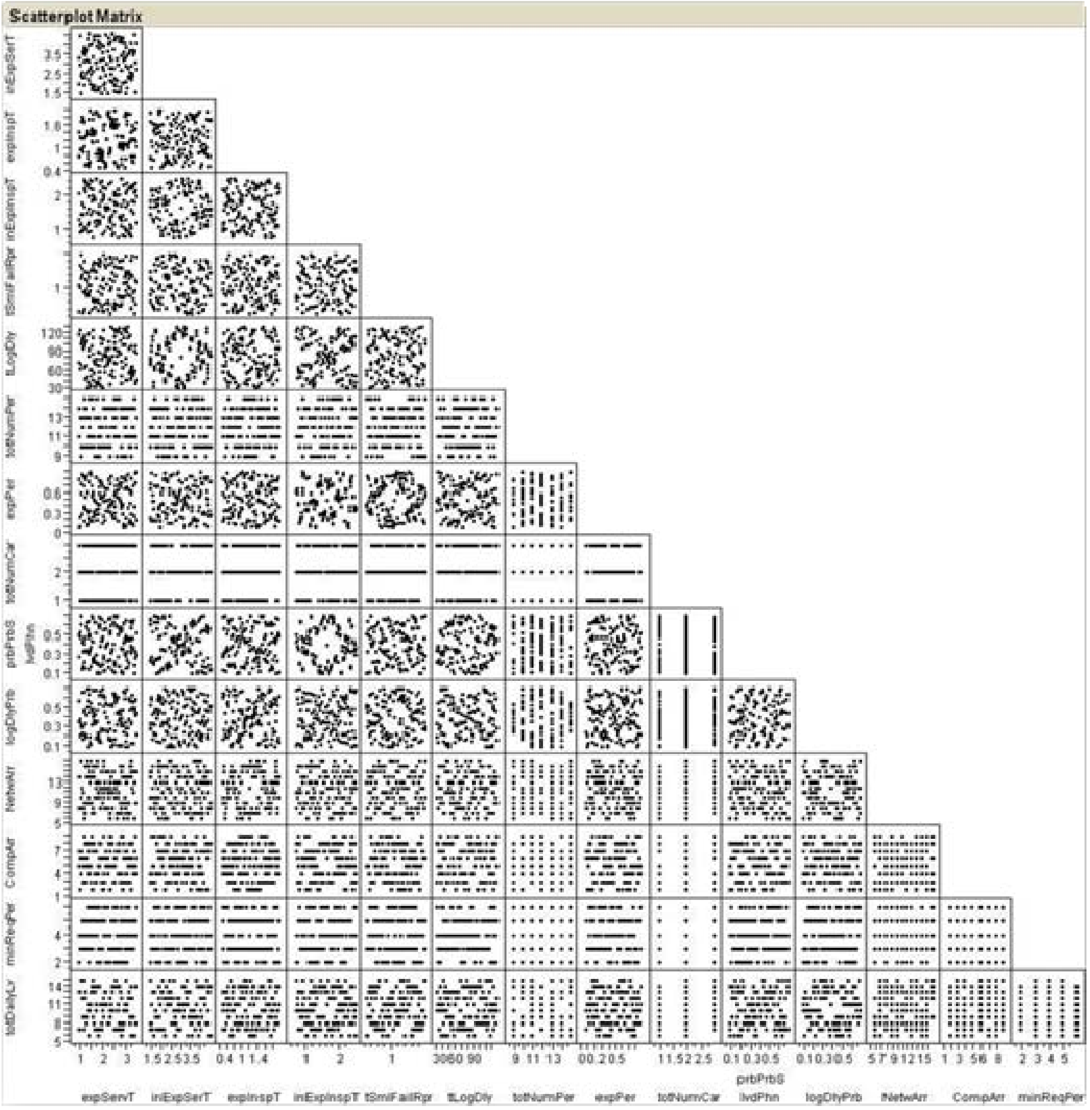


Figure 20. Scatter-plot matrix

E. SCENARIO REPLICATION

Replication is a must to deal with the stochastic characteristics of the model. The simulation cannot be run with just one replication unless the analyst is willing to make the assumption that the variability in the response is constant across all design points—an assumption that is clearly inappropriate for queuing systems. The output will change with each replication because of the different values obtained from the random number generator. Therefore, the simulation should be replicated by using the same design points many times to identify important factors better, and to quantify the variability in the responses.

In this simulation, 100 replications were used for each design point. Therefore, the simulation was run 12,900 (129 design points * 100 replications) times.

Now that the experimental design has been created, the next chapter presents the results of the analysis.

V. ANALYSIS AND RESULTS

This chapter explains the results obtained from the simulation runs. Here are the key elements that were used in the simulation.

- 15 input factors,
- 2 measures of effectiveness,
- 129 design points for 15 input factors,
- 100 replications,
- 12,900 simulation runs.

For the analysis part, the interactive, comprehensive, and highly visual statistical software, JMP 7.0, was used (SAS Institute Inc., 2007).

Several different models of the input/output relationships were fit using this software. A few are presented below, namely, the linear multiple regression analysis without interactions between the factors, regression with two-way interaction terms and quadratic effects, and finally non-parametric models called partition trees.

A. REGRESSION ANALYSIS

In this section, the analysis shown below is used to interpret the relationship between the input factors and the MOE values:

- Regression analysis, to explain the relationship stated above,
- R^2 values, to understand how much variability our factors can explain,
- Sorted parameters, to understand the importance order of the factors,
- Residual-by-predicted plot, to check the randomness,
- Interaction profiler to understand how the input factors interact.

Note that the regression models are fit using the average results for each design point, rather than the raw data, so R^2 represents how much variability in the mean of 100 replications can be explained by the model.

1. Mean Delay Time in System

a. Without Interactions

In this model, a stepwise regression analysis was done for the fifteen main input factors. The overall p-value of the regression is the probability of obtaining a relationship at least as strong as that observed purely by chance, assuming that no relationship exists. Typically, a p-value < 0.05 is used to identify "statistical significance." The p-value of the overall regression analysis for the first MOE, computer failures delay in the system, is less than 0.0001. It can also be seen from the actual-by-predicted plot in Figure 21. Therefore, it can be said that there is a strong relationship between the input variables and the response.

Furthermore, the R^2 value of 0.84 tells that 84% of the variability in the response variable is explained by the model. Although this is high, the slight curvature evident in the graph indicates that an even better regression model may exist.

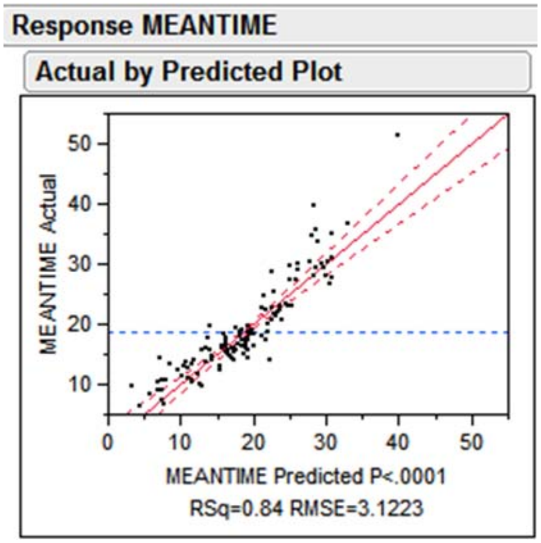
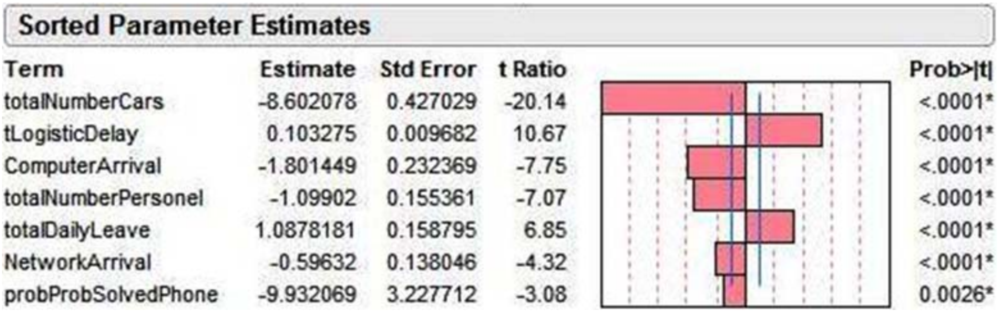


Figure 21. Actual by predicted plot for mean time in system

JMP also provides some other useful graphs to interpret the results. For example, it sorts the parameters by their importance. There are seven input factors that affect the output at the 0.95 confidence level. By looking at the chart in Figure 22, users can understand how important an input factor is to the result.



Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
totalNumberCars	-8.602078	0.427029	-20.14	<.0001*
tLogisticDelay	0.103275	0.009682	10.67	<.0001*
ComputerArrival	-1.801449	0.232369	-7.75	<.0001*
totalNumberPersonel	-1.09902	0.155361	-7.07	<.0001*
totalDailyLeave	1.0878181	0.158795	6.85	<.0001*
NetworkArrival	-0.59632	0.138046	-4.32	<.0001*
probProbSolvedPhone	-9.932069	3.227712	-3.08	0.0026*

Figure 22. Sorted parameters for mean time in system

The first and fourth most important factors are total number of the cars and personnel, respectively. Coefficients for both of these factors are negative, indicating that higher numbers of cars or personnel are associated with lower mean times in the system. Having enough cars is important for the repair jobs on a base. Transportation and going from one place to another is highly dependent on the number of cars. If there are not enough cars, it takes more time to get to the failure location due to waiting time for an available car, even if personnel are available to do the repair job. In short, the directions of these effects make sense.

As anticipated, the logistic delay time also has a large impact: it has the second-largest effect on the time in system due to being the biggest delay in the simulation. It takes 4–15 days, on average, for the logistics command to acquire the needed parts. Longer logistics delays have a negative impact on the system.

Another important factor is total daily leave. This parameter has a positive impact on the meantime in system, that is, total delay time increases as this factor increases. This factor explains how many excused daily leaves a technician can get for one-year period. This daily leave number is a maximum of 15 days per year, but usage is optional and highly dependent on the commander and the workload of the unit that the personnel belong to. It may also change from one unit to another.

Also, arrival rates for computer and network failures both play important roles on the delay time. Both of these factors show the inter-arrival times. Therefore, they have a negative impact on the MOE. The total time in system decreases when the inter-arrival time increases.

The first step taken when a problem emerges is calling the computer center and talking to a specialist to determine whether the problem is something that can be solved by giving instructions over the phone or by remote access. If the problem cannot be solved after this stage, then either the computer can be called to the center or personnel can be sent to repair the failure on site. Therefore, the probability of not solving the problem at this stage increases the total time in system. For such problems, both personnel and car availability become important.

Figure 23 presents the residual-by-predicted plot. As it is seen from the figure, there is a slight curvature in the plot. This may be due to the need for a more complex model that includes some interactions and quadratic effects. Therefore, both interactions and quadratic effects will be added, respectively, and the results will be discussed in the following section.

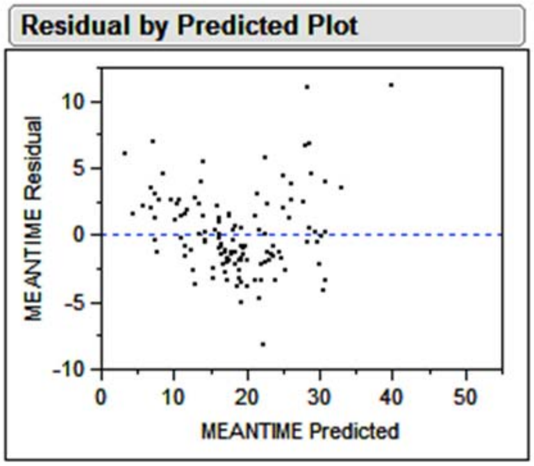


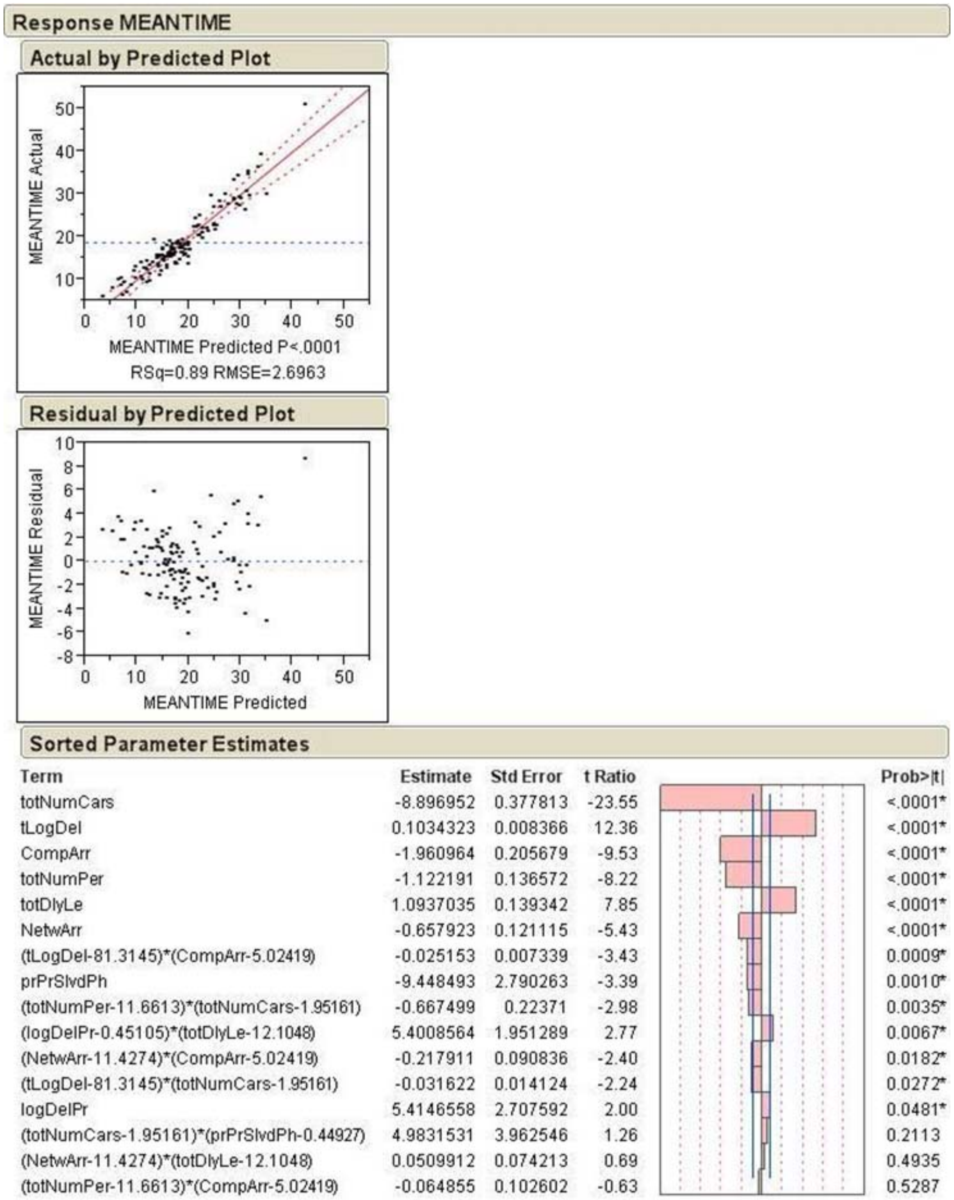
Figure 23. Residual-by-predicted plot for mean time in system

b. With Interactions

As the first step, two-way interactions are added to the model to attempt to solve the slight curvature problem in Figure 23. The regression model was fit by using stepwise regression, resulting in a model with eight significant main effects and eight interactions.

The model improved substantially and fit better after adding the two-way interaction for the selected main factors. Now there is less curvature in the residual-by-predicted plot as it is shown in the Figure 24. The p-value is still low and 0.0001. And it shows that there is a linear relationship between input factors and response variable. Furthermore, there is a noticeable increase in the R^2 value. It became 0.89 by an increase of 0.05. This means that 89% of the variability is explained by the model terms.

When the sorted parameters are inspected for both regressions, it can be realized that the important main factors are almost the same as the results obtained from the regression without interactions. However, since there are also some important interactions, the interaction profiler plot will now be examined to understand the interactions between the input factors better.



Response MEANTIME

Actual by Predicted Plot

Residual by Predicted Plot

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio		Prob>|t|
totNumCars	-8.896952	0.377813	-23.55		<.0001*
tLogDel	0.1034323	0.008366	12.36		<.0001*
CompArr	-1.960964	0.205679	-9.53		<.0001*
totNumPer	-1.122191	0.136572	-8.22		<.0001*
totDlyLe	1.0937035	0.139342	7.85		<.0001*
NetwArr	-0.657923	0.121115	-5.43		<.0001*
(tLogDel-81.3145)*(CompArr-5.02419)	-0.025153	0.007339	-3.43		0.0009*
prPrSlvdPh	-9.448493	2.790263	-3.39		0.0010*
(totNumPer-11.6613)*(totNumCars-1.95161)	-0.667499	0.22371	-2.98		0.0035*
(logDelPr-0.45105)*(totDlyLe-12.1048)	5.4008564	1.951289	2.77		0.0067*
(NetwArr-11.4274)*(CompArr-5.02419)	-0.217911	0.090836	-2.40		0.0182*
(tLogDel-81.3145)*(totNumCars-1.95161)	-0.031622	0.014124	-2.24		0.0272*
logDelPr	5.4146558	2.707592	2.00		0.0481*
(totNumCars-1.95161)*(prPrSlvdPh-0.44927)	4.9831531	3.962546	1.26		0.2113
(NetwArr-11.4274)*(totDlyLe-12.1048)	0.0509912	0.074213	0.69		0.4935
(totNumPer-11.6613)*(CompArr-5.02419)	-0.064855	0.102602	-0.63		0.5287

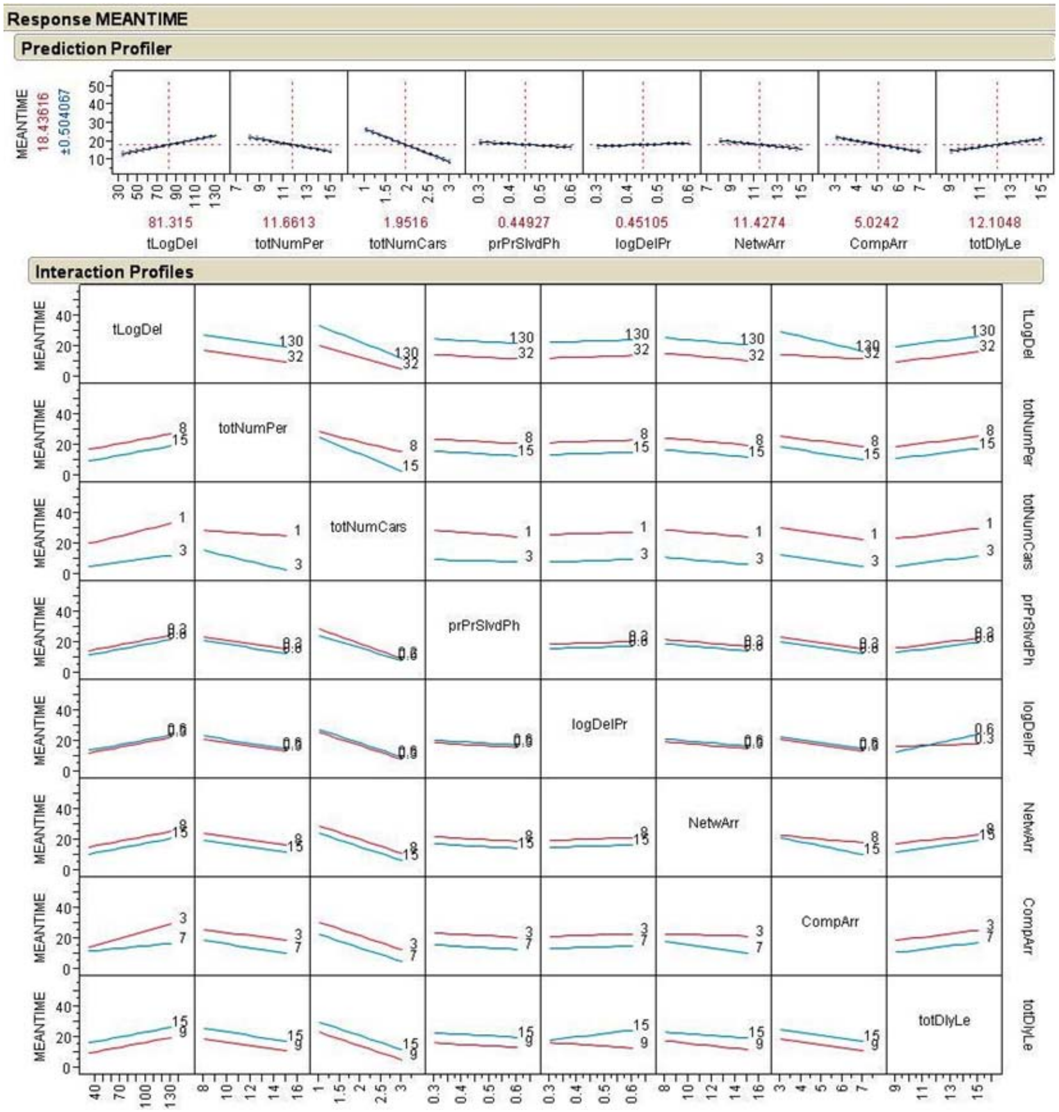
Figure 24. Regression analysis of mean time in system with the interactions

Figure 25 shows the interaction profiler. That graph shows the interaction between the input factors and their effects on changing situations.

The first remarkable interaction is the one between the logistic delay probability and computer arrival. For the logistic delay of 32 hours, an increase in the inter-arrival time for the computer failures does not change the mean time in the system. However, mean time in system dramatically decreases with the increase of inter-arrival time for the 130 hours of logistic delay time. This effect is very normal because when inter-arrival time is low (that is, failures arrive more frequently) and logistic delay time is high, part consumption increases.

Another interesting interaction is between the total number of cars and personnel. If there is only one car present for the personnel to use to get to repair locations, then the mean time in system is almost the same for both 8 and 15 personnel. The importance of personnel on the meantime in system becomes clear when the number of car increases to 3. As it can be seen from the profiler plot, the mean time is lower for 15 personnel at the level of 3 cars.

Another noteworthy interaction is between the probability of logistic delay and total daily leave. Here, total daily leave shows the number of days of excused leave each personnel is allowed. As it is mentioned earlier, personnel can use all of the optional 15 days of excused leave only at the discretion of the commander of that unit. As it is seen from the plot, there is no effect of the daily leave number on the mean delay time when the logistic delay probability is 0.3, that is, low. However, it becomes important when the logistic delay probability increases.



81.315 tLogDel

11.6613 totNumPer

1.9516 totNumCars

0.44927 prPrSlvdPh

0.45105 logDelPr

11.4274 NetwArr

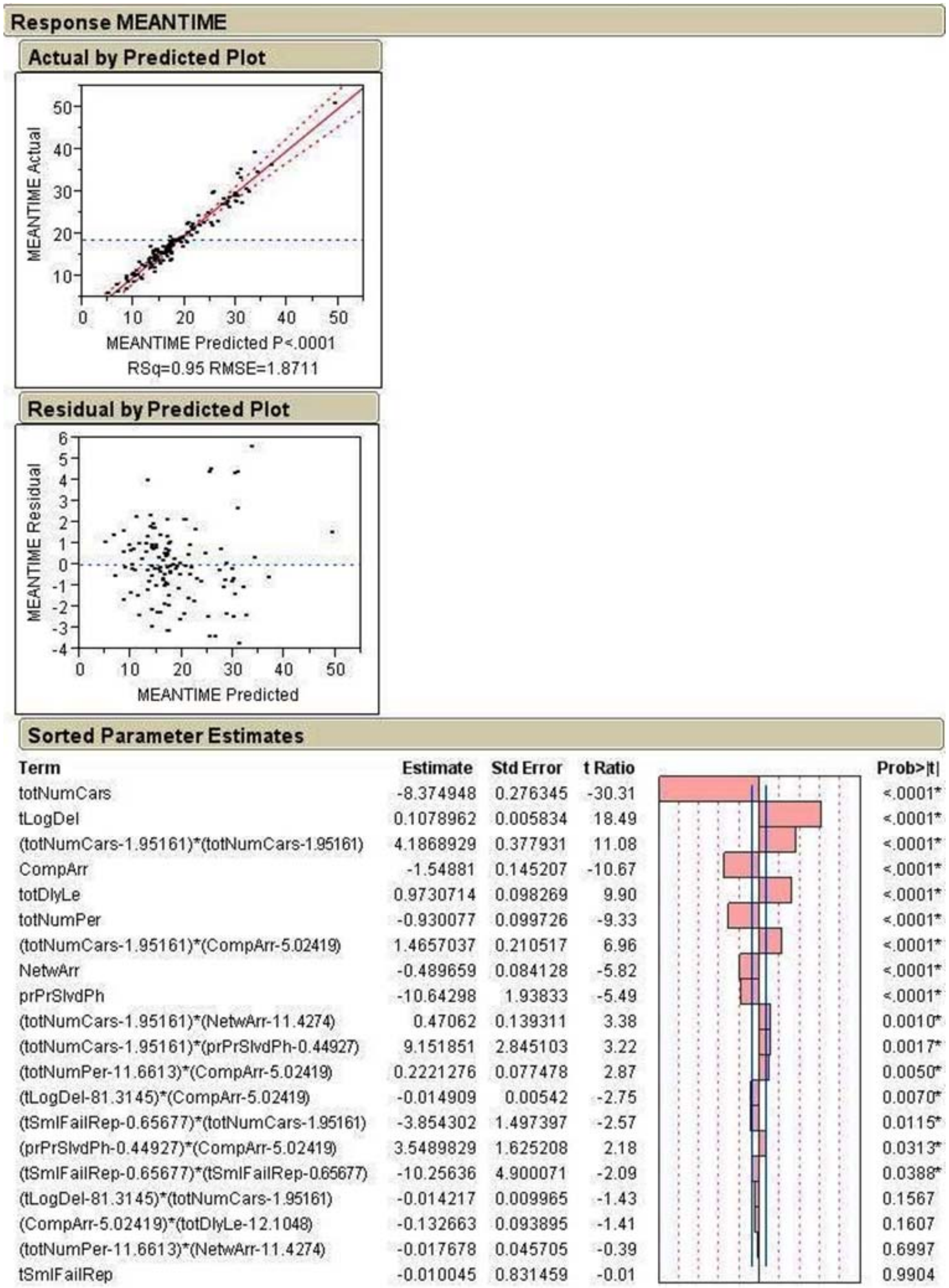
5.0242 CompArr

12.1048 totDlyLe

Figure 25. Interaction Profiler for mean time in system with interactions

c. With Interactions and Quadratic Effects

This time, both interactions and quadratic effects associated with the seven important factors are considered, and the model is created by using stepwise regression. The p-value is still 0.0001, which shows a significant relationship between the input factors and the response variable. The R^2 value increased from 0.89 to 0.95 by adding the quadratic effects (Figure 26). Thus, this model explains more variability in the response variable. Moreover, no pattern is observed from the residual-by-predicted plot—the two quadratic effects included in the model account for the curvature seen earlier. Note that the model could be simplified further by removing the three non-significant interaction terms; the non-significant main effect should remain in the model because this factor appears in some significant interactions.



Response MEANTIME

Actual by Predicted Plot

Residual by Predicted Plot

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
totNumCars	-8.374948	0.276345	-30.31	<.0001*
tLogDel	0.1078962	0.005834	18.49	<.0001*
(totNumCars-1.95161)*(totNumCars-1.95161)	4.1868929	0.377931	11.08	<.0001*
CompArr	-1.54881	0.145207	-10.67	<.0001*
totDlyLe	0.9730714	0.098269	9.90	<.0001*
totNumPer	-0.930077	0.099726	-9.33	<.0001*
(totNumCars-1.95161)*(CompArr-5.02419)	1.4657037	0.210517	6.96	<.0001*
NetwArr	-0.489659	0.084128	-5.82	<.0001*
prPrSlvdPh	-10.64298	1.93833	-5.49	<.0001*
(totNumCars-1.95161)*(NetwArr-11.4274)	0.47062	0.139311	3.38	0.0010*
(totNumCars-1.95161)*(prPrSlvdPh-0.44927)	9.151851	2.845103	3.22	0.0017*
(totNumPer-11.6613)*(CompArr-5.02419)	0.2221276	0.077478	2.87	0.0050*
(tLogDel-81.3145)*(CompArr-5.02419)	-0.014909	0.00542	-2.75	0.0070*
(tSmlFailRep-0.65677)*(totNumCars-1.95161)	-3.854302	1.497397	-2.57	0.0115*
(prPrSlvdPh-0.44927)*(CompArr-5.02419)	3.5489829	1.625208	2.18	0.0313*
(tSmlFailRep-0.65677)*(tSmlFailRep-0.65677)	-10.25636	4.900071	-2.09	0.0388*
(tLogDel-81.3145)*(totNumCars-1.95161)	-0.014217	0.009965	-1.43	0.1567
(CompArr-5.02419)*(totDlyLe-12.1048)	-0.132663	0.093895	-1.41	0.1607
(totNumPer-11.6613)*(NetwArr-11.4274)	-0.017678	0.045705	-0.39	0.6997
tSmlFailRep	-0.010045	0.831459	-0.01	0.9904

Figure 26. Regression analysis for mean time in system with interactions and quadratic effects

As seen from the Figure 27, the total number of cars has the most significant quadratic effect. It has a negative slope, and its slope between 1 and 2 is higher than the one between 2 and 3. Thus, the change in the mean time in system gets higher when we change the value of the number of cars between 1 and 2.

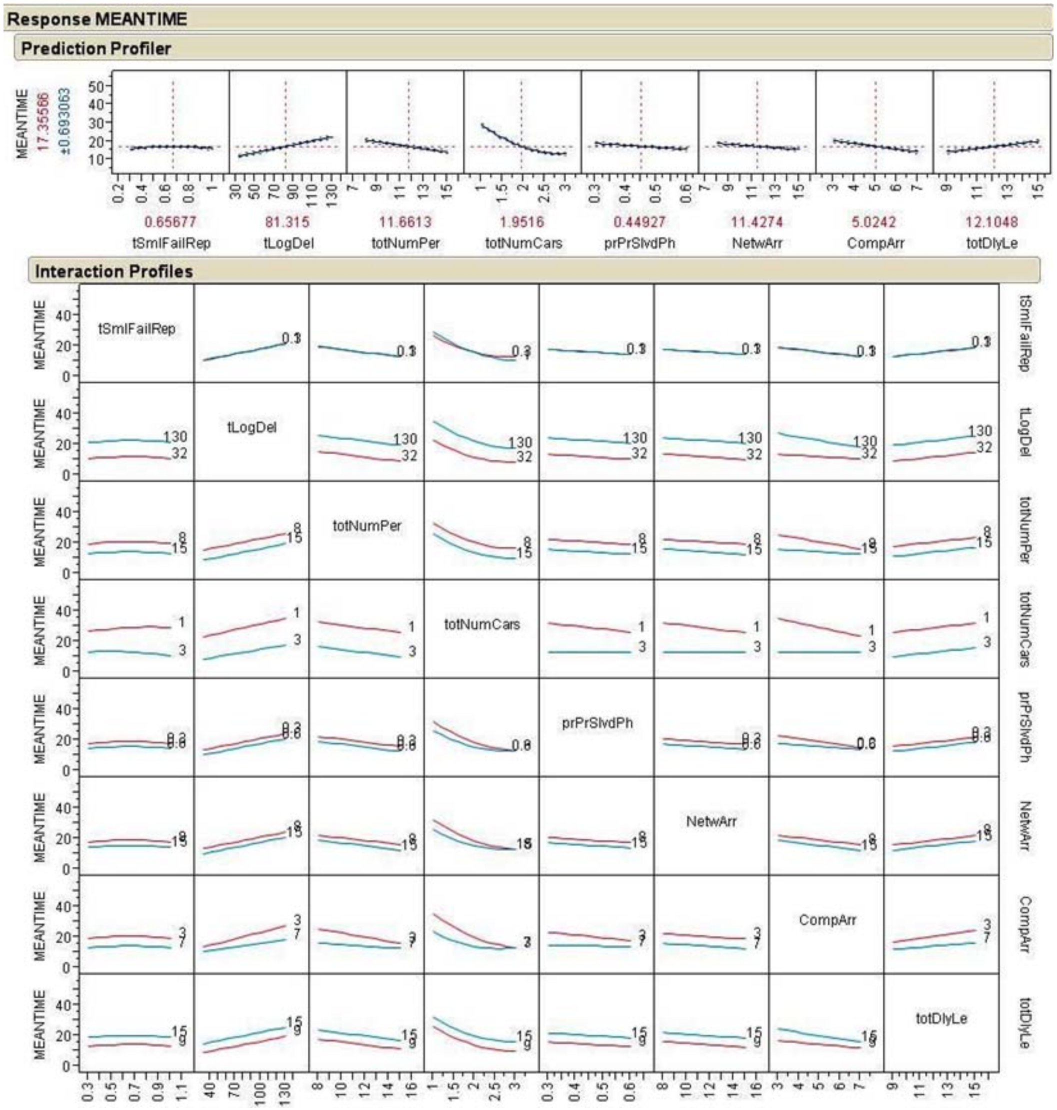


Figure 27. Interaction profiler for mean time in system with interactions and quadratic effects

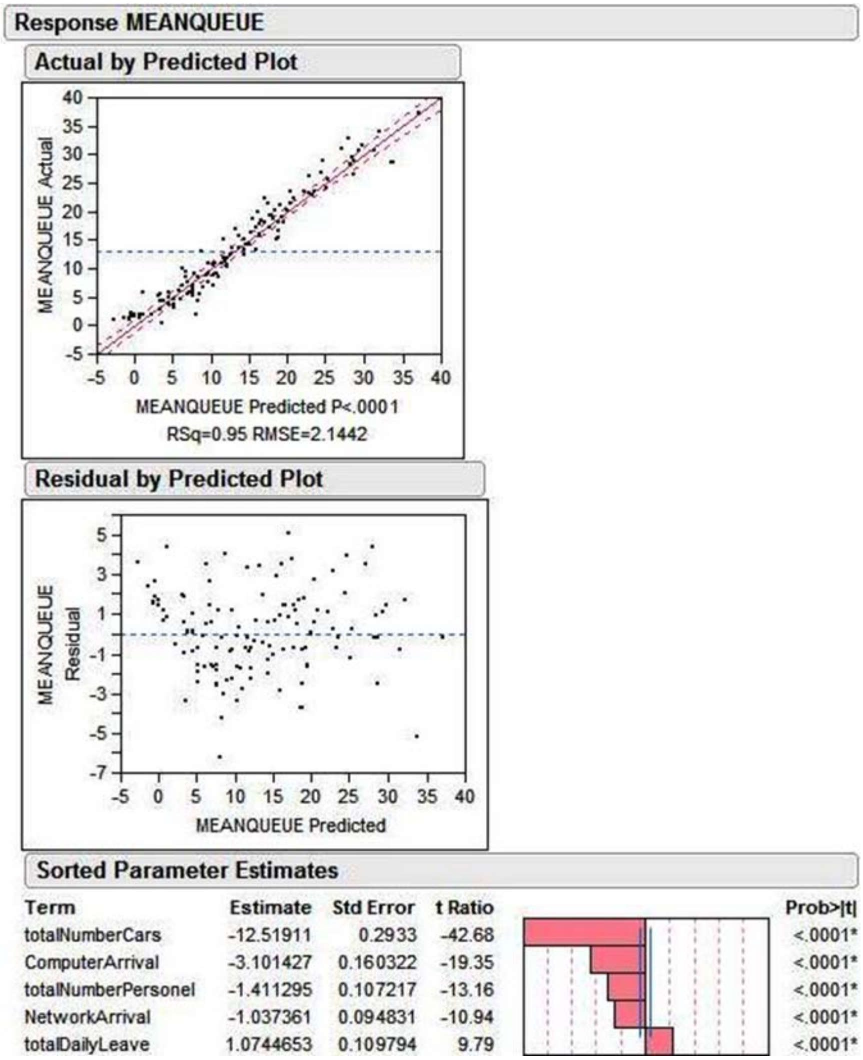
2. Mean Number of Failures Waiting in the Queue

a. Without Interactions

The p-value is small and the R^2 value is 0.95. The R^2 value of 95 percent is good enough to explain most of the variability in the response variable.

When the residual-by-predicted plot is observed in Figure 28, it can be seen that there is no particular pattern.

There are only five important input factors in this model. Again, the number of cars and personnel have beneficial effects on the MOE. The total number of cars parameter’s coefficient is the biggest of all. Thus, it has the maximum contribution on predictions.



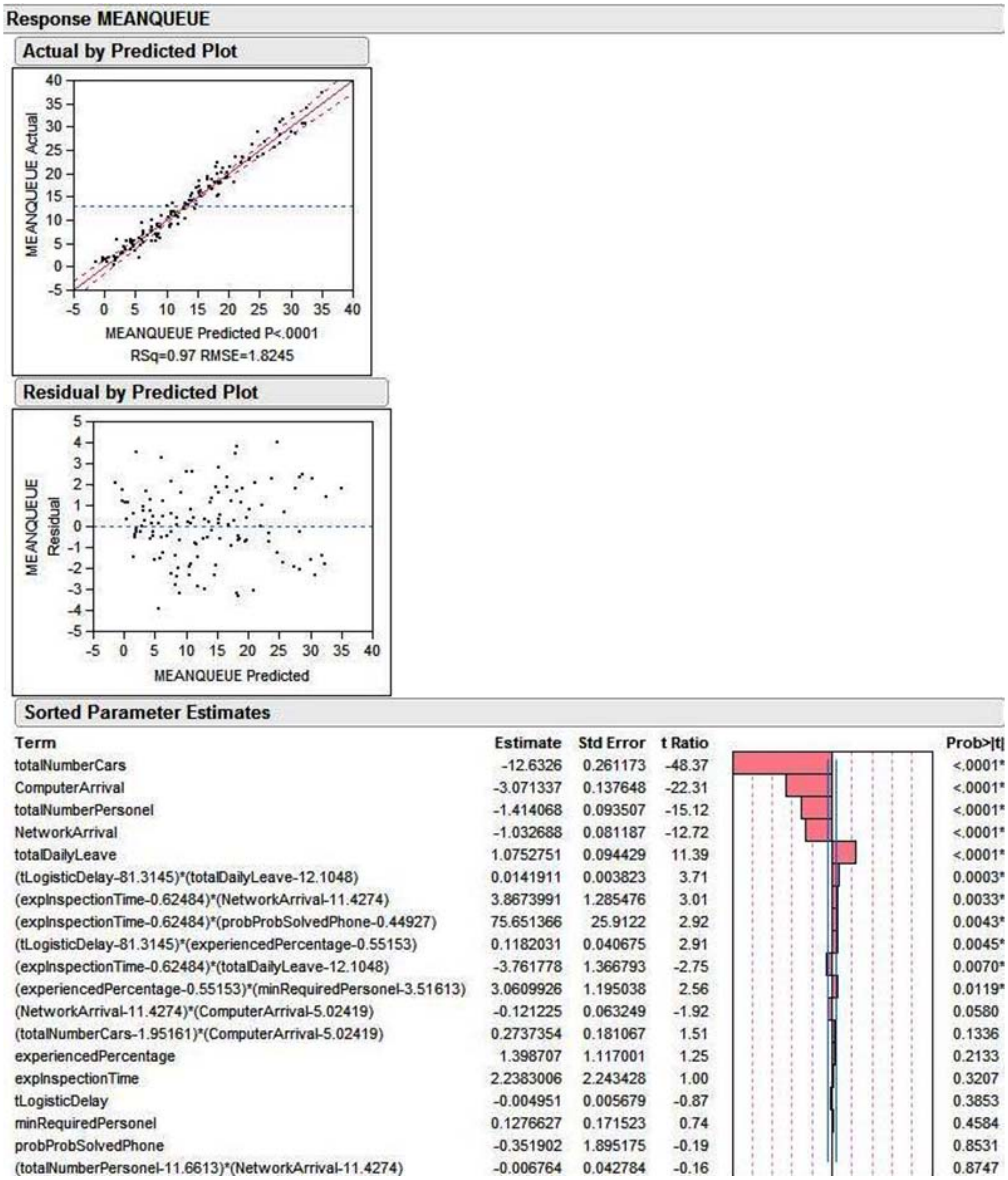
Term	Estimate	Std Error	t Ratio	Prob>|t|
totalNumberCars	-12.51911	0.2933	-42.68	<.0001*
ComputerArrival	-3.101427	0.160322	-19.35	<.0001*
totalNumberPersonel	-1.411295	0.107217	-13.16	<.0001*
NetworkArrival	-1.037361	0.094831	-10.94	<.0001*
totalDailyLeave	1.0744653	0.109794	9.79	<.0001*

Figure 28. Regression analysis for mean number of failures in queue

b. With Interactions

Adding interaction increased the R^2 value from 0.95 to 0.97. Therefore, it did not add much in explaining the response variable. On the contrary, it increased the complexity of the model. For this reason, it is not necessary to add the interactions to the model.

There is no pattern in the residual-by-predicted plot, as shown in Figure 29.



Response MEANQUEUE

Actual by Predicted Plot

Residual by Predicted Plot

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
totalNumberCars	-12.6326	0.261173	-48.37	<.0001*
ComputerArrival	-3.071337	0.137648	-22.31	<.0001*
totalNumberPersonel	-1.414068	0.093507	-15.12	<.0001*
NetworkArrival	-1.032688	0.081187	-12.72	<.0001*
totalDailyLeave	1.0752751	0.094429	11.39	<.0001*
(tLogisticDelay-81.3145)*(totalDailyLeave-12.1048)	0.0141911	0.003823	3.71	0.0003*
(expInspectionTime-0.62484)*(NetworkArrival-11.4274)	3.8673991	1.285476	3.01	0.0033*
(expInspectionTime-0.62484)*(probProbSolvedPhone-0.44927)	75.651366	25.9122	2.92	0.0043*
(tLogisticDelay-81.3145)*(experiencedPercentage-0.55153)	0.1182031	0.040675	2.91	0.0045*
(expInspectionTime-0.62484)*(totalDailyLeave-12.1048)	-3.761778	1.366793	-2.75	0.0070*
(experiencedPercentage-0.55153)*(minRequiredPersonel-3.51613)	3.0609926	1.195038	2.56	0.0119*
(NetworkArrival-11.4274)*(ComputerArrival-5.02419)	-0.121225	0.063249	-1.92	0.0580
(totalNumberCars-1.95161)*(ComputerArrival-5.02419)	0.2737354	0.181067	1.51	0.1336
experiencedPercentage	1.398707	1.117001	1.25	0.2133
expInspectionTime	2.2383006	2.243428	1.00	0.3207
tLogisticDelay	-0.004951	0.005679	-0.87	0.3853
minRequiredPersonel	0.1276627	0.171523	0.74	0.4584
probProbSolvedPhone	-0.351902	1.895175	-0.19	0.8531
(totalNumberPersonel-11.6613)*(NetworkArrival-11.4274)	-0.006764	0.042784	-0.16	0.8747

Figure 29. Regression analysis for mean number of failures in queue with interactions

B. REGRESSION TREES

Regression trees (sometimes called classification or partition trees) are another useful way of analyzing classification and regression problems. They are constructed using if-then statements; therefore understanding the interactions between the input factors is often easier when compared to regression analysis. Moreover, the threshold values provided by the regression tree analysis may provide better insight about “good” and “best” combinations of settings for the input factors than regression analysis. Therefore, regression analysis is good for understanding the important factors and how they contribute to explaining the response variable, while regression trees are good for giving actual numbers by providing the effects of those threshold values on the response variable.

Now the partition trees for both mean numbers of failed systems in the queue and meantime in system for those failed systems will be explained. JMP gives the user the freedom of choosing the number of splits. The user can see the increase in the R^2 value and decide whether further splits are necessary or not.

In the first partition tree, a 0.804 R^2 value is obtained in five splits; six or more splits do not add substantive increases to the R^2 value other than complicating the analysis. Different combinations may be observed when Figure 30 is inspected. For example, the most important factor is total number of cars. This factor was also important after the regression analysis, but now we can see its impacts on the response variable for under and above certain threshold values. For instance, when there are less than two cars, then the mean queue gets higher than 24 failed systems in queue. This is a high number of failed systems to wait in the queue. Therefore, it is necessary to have two or more cars to make the mean queue number reasonable. The mean can be decreased to approximately three failed systems, provided that there are more than three cars. However, if there are exactly two cars, it will be necessary to control the computer failure inter-arrival times. Actually, this is a hard factor to control. However, there are some solutions that might increase the computer inter-arrival time. For example, the failure rate of the parts may be decreased by acquiring better quality parts, or training the users might decrease user-related failures.

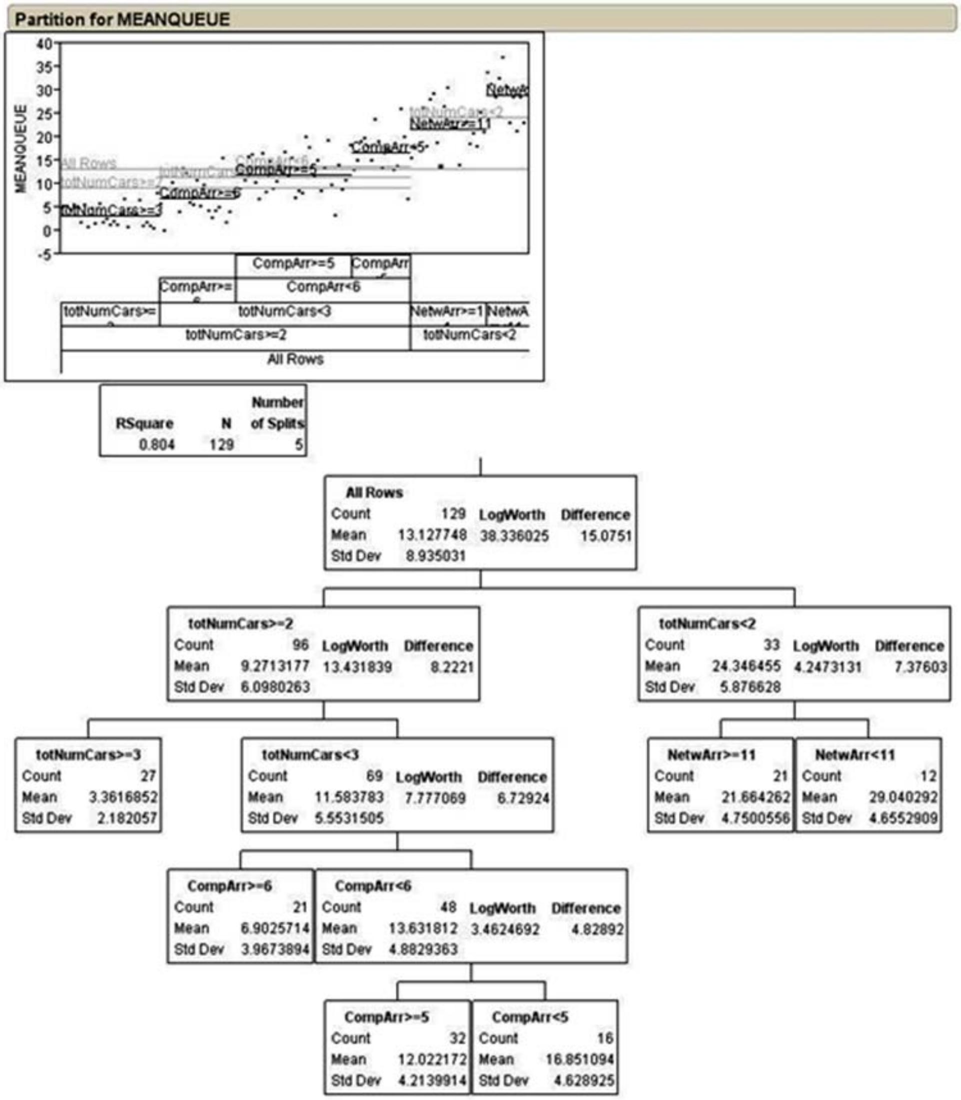
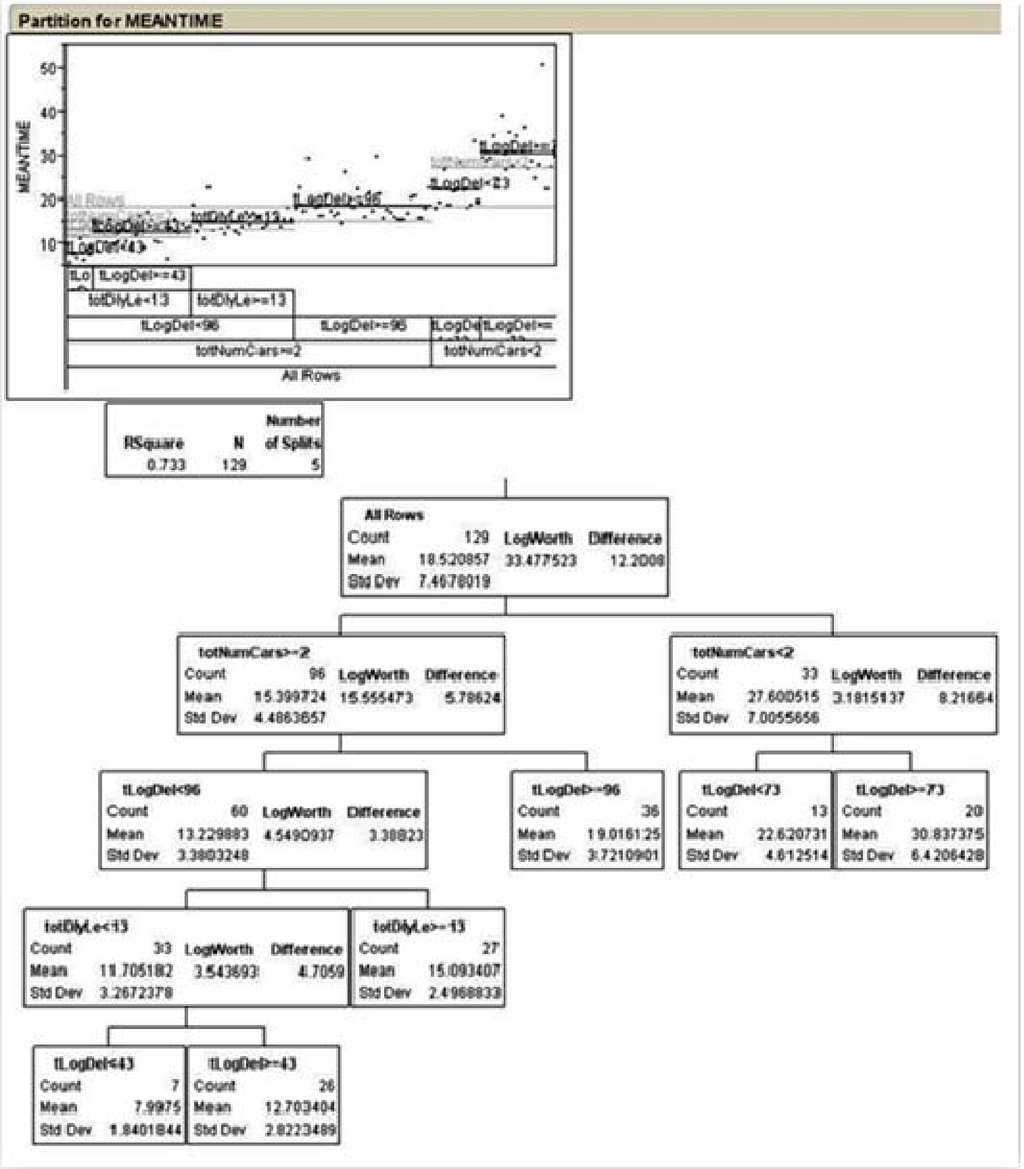


Figure 30. Partition tree for mean number of failures in queue

In the partition tree for mean time in the system, the R^2 value is 0.74 for five splits. Again, the different options can be observed from Figure 31. For example, if there are not more than two cars, then the mean time in the system gets higher than 27 hours. Logistic delay time is the most important factor after number of cars. Therefore, having more than two cars available all the time is necessary to get reasonable time values. For example, a 13-hour time in system value can be reached by having more than two cars and a logistic delay time that is less than 96 hours. To make it little better, total optional daily leave of personnel may be limited to 13 days. This limitation on the personnel daily leave reduces the mean time system to 11 hours.



RSquare	N	Number of Splits
0.733	129	5

All Rows
Count 129 LogWorth 33.477523 Difference 12.2008
Mean 18.520857
Std Dev 7.4678019

totNumCars>=2
Count 96 LogWorth 15.555473 Difference 5.78624
Mean 15.399724
Std Dev 4.4863657

totNumCars<2
Count 33 LogWorth 3.1815137 Difference 8.21664
Mean 27.600515
Std Dev 7.0055656

tLogDel<96
Count 60 LogWorth 4.5490937 Difference 3.38823
Mean 13.229883
Std Dev 3.3803248

tLogDel>=96
Count 36
Mean 19.016125
Std Dev 3.7210901

tLogDel<73
Count 13
Mean 22.620731
Std Dev 4.612514

tLogDel>=73
Count 20
Mean 30.837375
Std Dev 6.4206428

totDlyLe<13
Count 33 LogWorth 3.543693 Difference 4.7059
Mean 11.705182
Std Dev 3.2672378

totDlyLe>=13
Count 27
Mean 15.093407
Std Dev 2.4968833

tLogDel<43
Count 7
Mean 7.9975
Std Dev 1.8401844

tLogDel>=43
Count 26
Mean 12.703404
Std Dev 2.8223489

Figure 31. Partition tree for mean waiting time in system

The regression and regression tree analysis were done to understand the effects of the input factors on the response variables: mean time in system, and mean down systems in the queue. As the result of the analysis, several factors were determined to be important. The most important factor is the number of the cars. As mentioned earlier, cars are used to get the computer specialists to the failure locations. If the number of cars is not enough, then personnel must wait for the next available car. Therefore, the first action taken to improve the computer center's performance may be increasing the number of cars to a reasonable level. The importance of regression tree may be understood here, because the result of that analysis gave us the threshold value for number of cars. It should be three or more for better results. Since three is the maximum number of cars in our experiment, this suggests that increasing the number further might be even more beneficial. This could be assessed after running another experiment.

Even though the total number of personnel did not appear in splits in the regression tree analysis, it did show up as an important factor in the results of the stepwise regression analysis. The number of experienced personnel (decided by the experience percentage) was expected to be an important factor at the beginning of the simulation; however, it was not included in either the regression analysis or the regression tree. This may be due to not including the situations where experience would be really important for resolving the issue in short times. For example, an out-of-the-ordinary network or computer system failure may be an example of this situation. Therefore, every personnel either experienced or inexperienced may be good at repairing the regular failures for which the repair steps are well known.

Also, the inter-arrival time for computer and network arrivals are significant. As it is mentioned earlier, these factors may be hard to control. However, it is not impossible to control them. For example, these values can be increased by acquiring high quality components. Furthermore, giving training to the users about how to use the systems more efficiently may affect the inter-arrival times.

Logistic delay time is another important factor, especially for the mean time in system. This delay occurs due to lacking in parts to repair the systems. It can be thought that acquiring more parts and making them ready for use all the time may solve this problem. Indeed, this is one possible solution to this logistic delay problem, but logistic command does not want to buy more parts than needed. The reason for this is the parts quickly become outdated as computer technology develops. Therefore, it becomes important to decide the part needs for a one-year period. This can be accomplished by adding another module to this study.

Even though the parameters used in this thesis mostly depend on real data, the assumptions in Chapter III should be assessed before using this study to take actions for the computer specialist NCOs' problem in the Turkish Air Force bases. It may also be good to update the time distributions and other parameters based on real data obtained from a specific base.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

In this thesis, a simulation tool to address the duties of the computer specialist NCOs on a base was designed. The main idea was to show that personnel increase is not the only solution for problems in a system. The handling of network and computer jobs in a Turkish Air Force Base proved this point. This study also identified those factors that have the most significant effects on the computer specialist's jobs, assuming that the assumptions and distributions in the simulation capture the essential characteristics of the real system. However, as it is stated earlier, there may be a need to update the time and other distribution values to obtain the most recent parameters and thus to get more accurate results before using this study for further analysis.

The methodologies used in this work were event graphs and discrete-event simulation techniques; the simulation tools, Simkit and Viskit; NOLH for an effective design; and, finally, the statistical analysis software, JMP, to make the analysis.

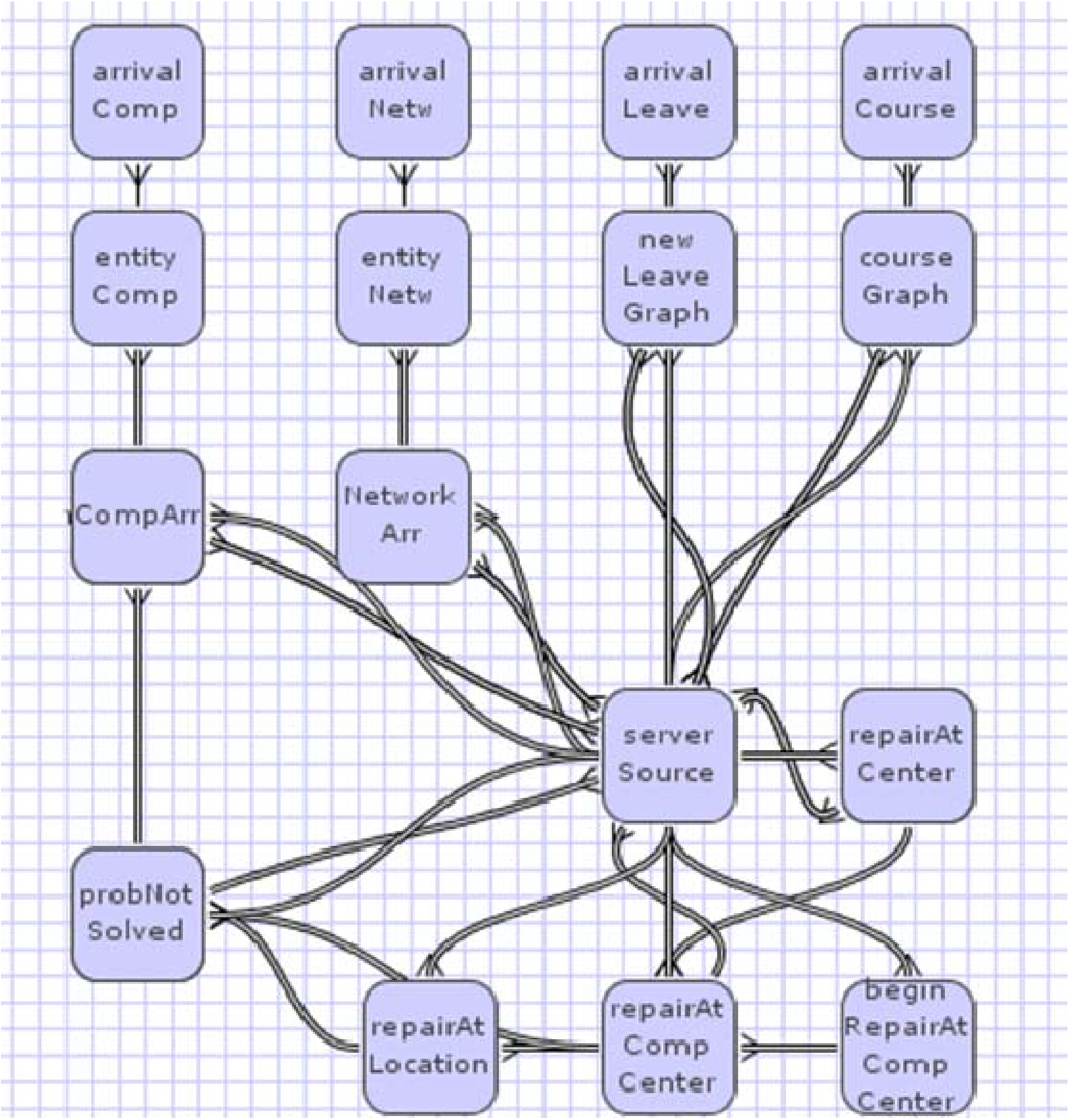
At the beginning of the study, 15 input factors were determined to be of interest. These factors were explained in detail in Chapter IV. After the simulation experiment was run and the analysis was conducted by using the JMP program, the importance of these factors in explaining our MOEs was revealed. Not all of the factors are equally important.

Generally, over the range of input factor levels for this experiment, the key determinants of performance are the number of personnel and the availability of cars; total daily leave; means of network and computer arrivals; and the logistic delay time. How these factors affect the MOEs is shown and explained in the analysis chapter. The personnel were divided in two groups—experienced or inexperienced—based on the experience percentage defined in the design. Experience level was expected to be more important than it turned out to be at the end of the simulation. However, it did not have a significant effect on either of the MOEs. The reason for this may be that the time values which differentiate the experienced and inexperienced personnel from each other are too close to one another. Therefore, they did not have an effect on the MOEs. Alternatively, there may be some situations that were not integrated into the simulation that truly would require more experience and knowledge to understand and solve quickly. Hence, we cannot conclude that experience level is not an important factor—this finding may be valid for only this model.

The results do show that increasing the staff is not the only solution for this particular research. There are some other factors that can be played with to decrease the mean number in queue and time in the system. Moreover, the Turkish Air Force can take this thesis as a basis to solve the personnel lacking issues in Air Force Base Computer Centers. As mentioned earlier, increasing the number of personnel is not the only solution for improvement. There are also some other factors that play a critical role on solving the staff problem. For instance, the number of cars is a key factor and has great effect on MOEs. Increasing the reliability and quality of computer and network systems and parts can decrease the number of failures and may result in less need for personnel for repairs. Logistic delay time can be decreased by acquiring more parts for repairs, although this .may not be cost effective since parts may easily be outdated because of developing computer technology. Therefore, another study should be made to decide the number of needed parts to fix most of the problems over one year period. Another factor is the probability of solving the problem remotely. This can potentially be improved by sending personnel to courses to increase their experience level, resulting in a workforce capable of solving more problems by connecting to the malfunctioning system remotely. Finally, not allowing personnel to use all of the optional 15 days of daily excuse may be considered as another option, and can be applied if needed.

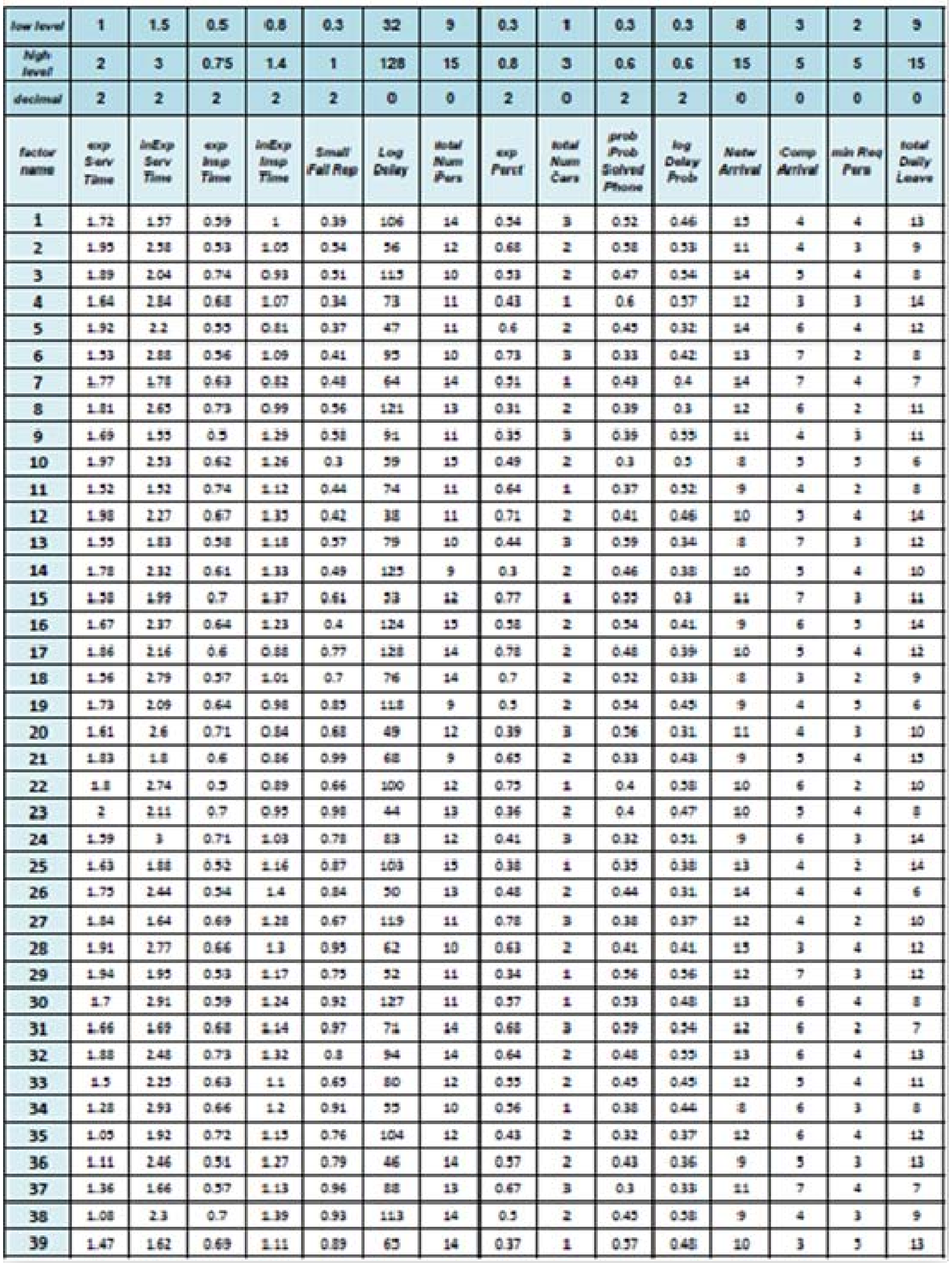
Identifying effective factors and showing how they can help solve the problem is explained above, and the same approach can be used if different factor ranges or distributions are of interest, or if the model is enhanced to relax or remove some of the assumptions. However, a trade-off study for the different setups of the input factors may be done as a future study to this thesis before taking some action. The experimental design can be used to identify promising alternatives, but the costs of these alternatives should be compared to come up with the best solution.

APPENDIX A: THESIS ASSEMBLY

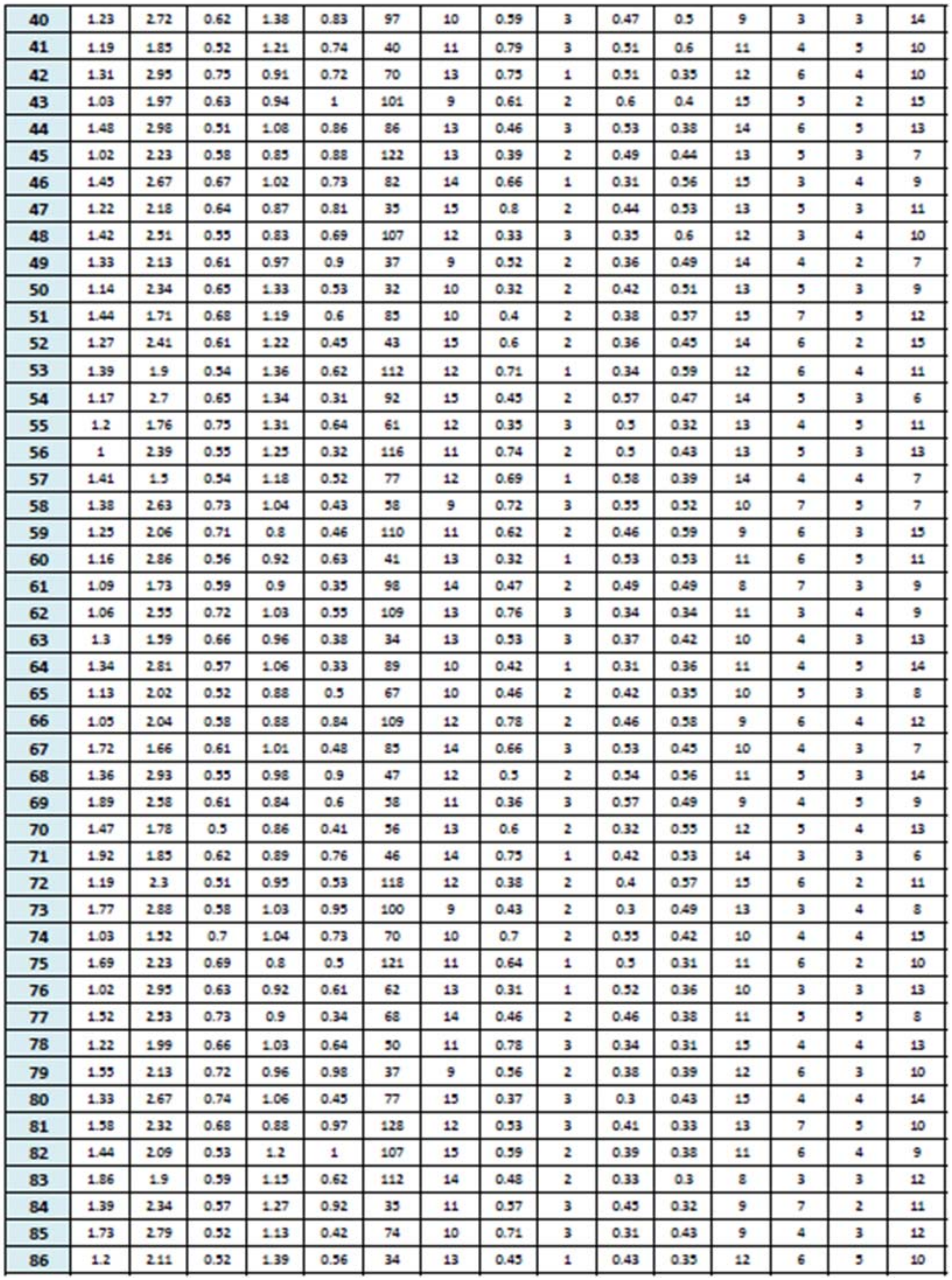


THIS PAGE INTENTIONALLY LEFT BLANK

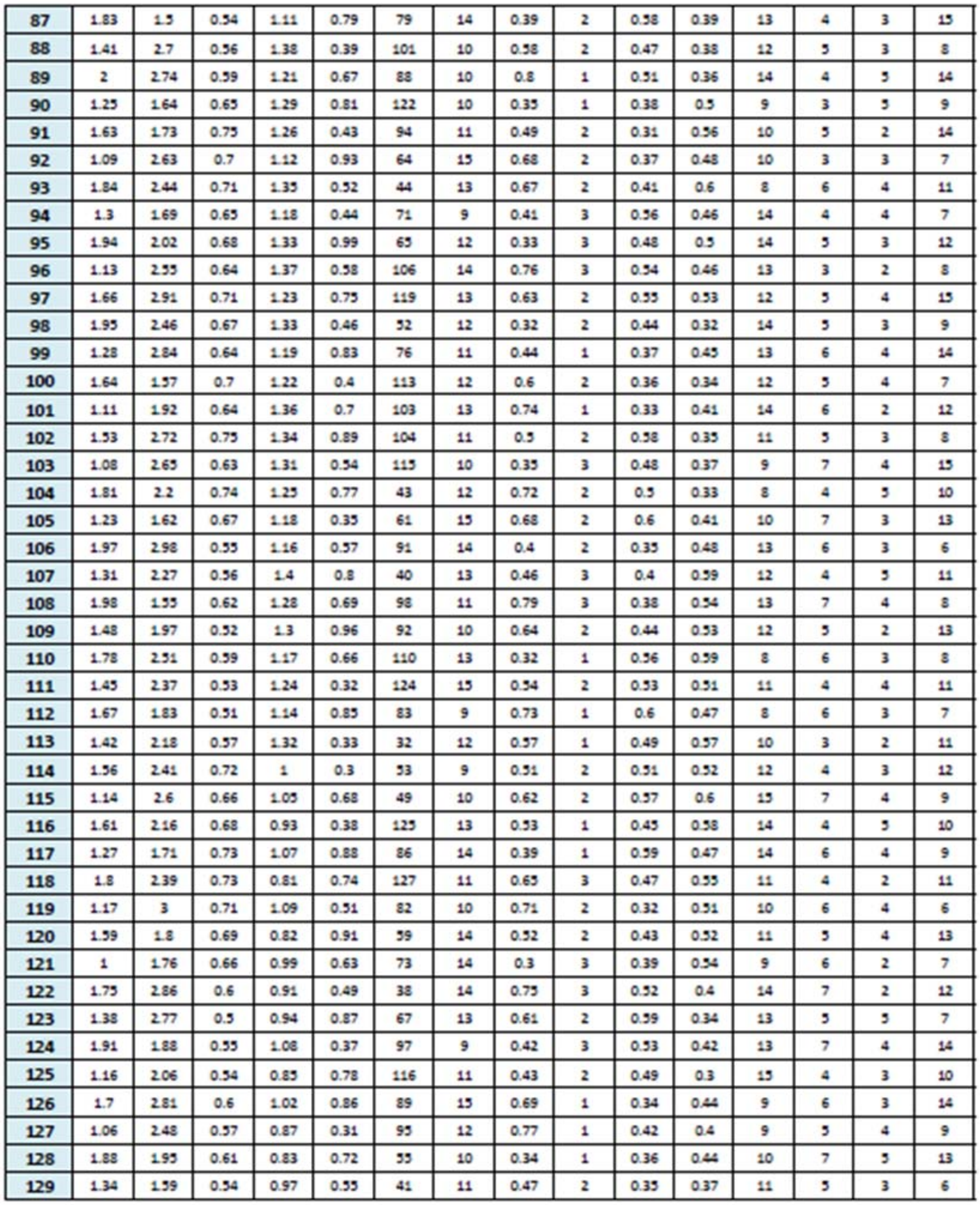
APPENDIX B: EXPERIMENTAL DESIGN



low level	1	1.5	0.5	0.8	0.3	32	9	0.3	1	0.3	0.3	8	3	2	9
high level	2	3	0.75	1.4	1	128	15	0.8	3	0.6	0.6	15	5	5	15
decimal	2	2	2	2	2	0	0	2	0	2	2	0	0	0	0
factor name	exp Serv Time	inExp Serv Time	exp Insp Time	inExp Insp Time	Small Fail Rep	Log Delay	total Num Pers	exp Perct	total Num Cars	prob Prob Solved Phone	log Delay Prob	Netw Arrival	Comp Arrival	min Req Pers	total Daily Leave
1	1.72	1.57	0.59	1	0.39	106	14	0.54	3	0.52	0.46	15	4	4	13
2	1.95	2.58	0.53	1.05	0.54	56	12	0.68	2	0.58	0.53	11	4	3	9
3	1.89	2.04	0.74	0.93	0.51	115	10	0.53	2	0.47	0.54	14	5	4	8
4	1.64	2.84	0.68	1.07	0.34	73	11	0.43	1	0.6	0.57	12	3	3	14
5	1.92	2.2	0.55	0.81	0.37	47	11	0.6	2	0.45	0.32	14	6	4	12
6	1.53	2.88	0.56	1.09	0.41	95	10	0.73	3	0.33	0.42	13	7	2	8
7	1.77	1.78	0.63	0.82	0.48	64	14	0.51	1	0.43	0.4	14	7	4	7
8	1.81	2.65	0.73	0.99	0.56	121	13	0.31	2	0.39	0.3	12	6	2	11
9	1.69	1.55	0.5	1.29	0.58	91	11	0.35	3	0.39	0.55	11	4	3	11
10	1.97	2.53	0.62	1.26	0.3	59	15	0.49	2	0.3	0.5	8	5	5	6
11	1.52	1.52	0.74	1.12	0.44	74	11	0.64	1	0.37	0.52	9	4	2	8
12	1.98	2.27	0.67	1.35	0.42	38	11	0.71	2	0.41	0.46	10	5	4	14
13	1.55	1.83	0.58	1.18	0.57	79	10	0.44	3	0.59	0.34	8	7	3	12
14	1.78	2.32	0.61	1.33	0.49	125	9	0.3	2	0.46	0.38	10	5	4	10
15	1.58	1.99	0.7	1.37	0.61	53	12	0.77	1	0.55	0.3	11	7	3	11
16	1.67	2.37	0.64	1.23	0.4	124	15	0.58	2	0.54	0.41	9	6	5	14
17	1.86	2.16	0.6	0.88	0.77	128	14	0.78	2	0.48	0.39	10	5	4	12
18	1.56	2.79	0.57	1.01	0.7	76	14	0.7	2	0.52	0.33	8	3	2	9
19	1.73	2.09	0.64	0.98	0.85	118	9	0.5	2	0.54	0.45	9	4	5	6
20	1.61	2.6	0.71	0.84	0.68	49	12	0.39	3	0.56	0.31	11	4	3	10
21	1.83	1.8	0.6	0.86	0.99	68	9	0.65	2	0.33	0.43	9	5	4	15
22	1.8	2.74	0.5	0.89	0.66	100	12	0.75	1	0.4	0.58	10	6	2	10
23	2	2.11	0.7	0.95	0.98	44	13	0.36	2	0.4	0.47	10	5	4	8
24	1.59	3	0.71	1.03	0.78	83	12	0.41	3	0.32	0.51	9	6	3	14
25	1.63	1.88	0.52	1.16	0.87	103	15	0.38	1	0.35	0.38	13	4	2	14
26	1.75	2.44	0.54	1.4	0.84	50	13	0.48	2	0.44	0.31	14	4	4	6
27	1.84	1.64	0.69	1.28	0.67	119	11	0.78	3	0.38	0.37	12	4	2	10
28	1.91	2.77	0.66	1.3	0.95	62	10	0.63	2	0.41	0.41	15	3	4	12
29	1.94	1.95	0.53	1.17	0.75	52	11	0.34	1	0.56	0.56	12	7	3	12
30	1.7	2.91	0.59	1.24	0.92	127	11	0.57	1	0.53	0.48	13	6	4	8
31	1.66	1.69	0.68	1.14	0.97	71	14	0.68	3	0.59	0.54	12	6	2	7
32	1.88	2.48	0.73	1.32	0.8	94	14	0.64	2	0.48	0.55	13	6	4	13
33	1.5	2.25	0.63	1.1	0.65	80	12	0.55	2	0.45	0.45	12	5	4	11
34	1.28	2.93	0.66	1.2	0.91	55	10	0.56	1	0.38	0.44	8	6	3	8
35	1.05	1.92	0.72	1.15	0.76	104	12	0.43	2	0.32	0.37	12	6	4	12
36	1.11	2.46	0.51	1.27	0.79	46	14	0.57	2	0.43	0.36	9	5	3	13
37	1.36	1.66	0.57	1.13	0.96	88	13	0.67	3	0.3	0.33	11	7	4	7
38	1.08	2.3	0.7	1.39	0.93	113	14	0.5	2	0.45	0.58	9	4	3	9
39	1.47	1.62	0.69	1.11	0.89	65	14	0.37	1	0.57	0.48	10	3	5	13



40	1.23	2.72	0.62	1.38	0.83	97	10	0.59	3	0.47	0.5	9	3	3	14
41	1.19	1.85	0.52	1.21	0.74	40	11	0.79	3	0.51	0.6	11	4	5	10
42	1.31	2.95	0.75	0.91	0.72	70	13	0.75	1	0.51	0.35	12	6	4	10
43	1.03	1.97	0.63	0.94	1	101	9	0.61	2	0.6	0.4	15	5	2	15
44	1.48	2.98	0.51	1.08	0.86	86	13	0.46	3	0.53	0.38	14	6	5	13
45	1.02	2.23	0.58	0.85	0.88	122	13	0.39	2	0.49	0.44	13	5	3	7
46	1.45	2.67	0.67	1.02	0.73	82	14	0.66	1	0.31	0.56	15	3	4	9
47	1.22	2.18	0.64	0.87	0.81	35	15	0.8	2	0.44	0.53	13	5	3	11
48	1.42	2.51	0.55	0.83	0.69	107	12	0.33	3	0.35	0.6	12	3	4	10
49	1.33	2.13	0.61	0.97	0.9	37	9	0.52	2	0.36	0.49	14	4	2	7
50	1.14	2.34	0.65	1.33	0.53	32	10	0.32	2	0.42	0.51	13	5	3	9
51	1.44	1.71	0.68	1.19	0.6	85	10	0.4	2	0.38	0.57	15	7	5	12
52	1.27	2.41	0.61	1.22	0.45	43	15	0.6	2	0.36	0.45	14	6	2	15
53	1.39	1.9	0.54	1.36	0.62	112	12	0.71	1	0.34	0.59	12	6	4	11
54	1.17	2.7	0.65	1.34	0.31	92	15	0.45	2	0.57	0.47	14	5	3	6
55	1.2	1.76	0.75	1.31	0.64	61	12	0.35	3	0.5	0.32	13	4	5	11
56	1	2.39	0.55	1.25	0.32	116	11	0.74	2	0.5	0.43	13	5	3	13
57	1.41	1.5	0.54	1.18	0.52	77	12	0.69	1	0.58	0.39	14	4	4	7
58	1.38	2.63	0.73	1.04	0.43	58	9	0.72	3	0.55	0.52	10	7	5	7
59	1.25	2.06	0.71	0.8	0.46	110	11	0.62	2	0.46	0.59	9	6	3	15
60	1.16	2.86	0.56	0.92	0.63	41	13	0.32	1	0.53	0.53	11	6	5	11
61	1.09	1.73	0.59	0.9	0.35	98	14	0.47	2	0.49	0.49	8	7	3	9
62	1.06	2.55	0.72	1.03	0.55	109	13	0.76	3	0.34	0.34	11	3	4	9
63	1.3	1.59	0.66	0.96	0.38	34	13	0.53	3	0.37	0.42	10	4	3	13
64	1.34	2.81	0.57	1.06	0.33	89	10	0.42	1	0.31	0.36	11	4	5	14
65	1.13	2.02	0.52	0.88	0.5	67	10	0.46	2	0.42	0.35	10	5	3	8
66	1.05	2.04	0.58	0.88	0.84	109	12	0.78	2	0.46	0.58	9	6	4	12
67	1.72	1.66	0.61	1.01	0.48	85	14	0.66	3	0.53	0.45	10	4	3	7
68	1.36	2.93	0.55	0.98	0.9	47	12	0.5	2	0.54	0.56	11	5	3	14
69	1.89	2.58	0.61	0.84	0.6	58	11	0.36	3	0.57	0.49	9	4	5	9
70	1.47	1.78	0.5	0.86	0.41	56	13	0.6	2	0.32	0.55	12	5	4	13
71	1.92	1.85	0.62	0.89	0.76	46	14	0.75	1	0.42	0.53	14	3	3	6
72	1.19	2.3	0.51	0.95	0.53	118	12	0.38	2	0.4	0.57	15	6	2	11
73	1.77	2.88	0.58	1.03	0.95	100	9	0.43	2	0.3	0.49	13	3	4	8
74	1.03	1.52	0.7	1.04	0.73	70	10	0.7	2	0.55	0.42	10	4	4	15
75	1.69	2.23	0.69	0.8	0.5	121	11	0.64	1	0.5	0.31	11	6	2	10
76	1.02	2.95	0.63	0.92	0.61	62	13	0.31	1	0.52	0.36	10	3	3	13
77	1.52	2.53	0.73	0.9	0.34	68	14	0.46	2	0.46	0.38	11	5	5	8
78	1.22	1.99	0.66	1.03	0.64	50	11	0.78	3	0.34	0.31	15	4	4	13
79	1.55	2.13	0.72	0.96	0.98	37	9	0.56	2	0.38	0.39	12	6	3	10
80	1.33	2.67	0.74	1.06	0.45	77	15	0.37	3	0.3	0.43	15	4	4	14
81	1.58	2.32	0.68	0.88	0.97	128	12	0.53	3	0.41	0.33	13	7	5	10
82	1.44	2.09	0.53	1.2	1	107	15	0.59	2	0.39	0.38	11	6	4	9
83	1.86	1.9	0.59	1.15	0.62	112	14	0.48	2	0.33	0.3	8	3	3	12
84	1.39	2.34	0.57	1.27	0.92	35	11	0.57	3	0.45	0.32	9	7	2	11
85	1.73	2.79	0.52	1.13	0.42	74	10	0.71	3	0.31	0.43	9	4	3	12
86	1.2	2.11	0.52	1.39	0.56	34	13	0.45	1	0.43	0.35	12	6	5	10



87	1.83	1.5	0.54	1.11	0.79	79	14	0.39	2	0.58	0.39	13	4	3	15
88	1.41	2.7	0.56	1.38	0.39	101	10	0.58	2	0.47	0.38	12	5	3	8
89	2	2.74	0.59	1.21	0.67	88	10	0.8	1	0.51	0.36	14	4	5	14
90	1.25	1.64	0.65	1.29	0.81	122	10	0.35	1	0.38	0.5	9	3	5	9
91	1.63	1.73	0.75	1.26	0.43	94	11	0.49	2	0.31	0.56	10	5	2	14
92	1.09	2.63	0.7	1.12	0.93	64	15	0.68	2	0.37	0.48	10	3	3	7
93	1.84	2.44	0.71	1.35	0.52	44	13	0.67	2	0.41	0.6	8	6	4	11
94	1.3	1.69	0.65	1.18	0.44	71	9	0.41	3	0.56	0.46	14	4	4	7
95	1.94	2.02	0.68	1.33	0.99	65	12	0.33	3	0.48	0.5	14	5	3	12
96	1.13	2.55	0.64	1.37	0.58	106	14	0.76	3	0.54	0.46	13	3	2	8
97	1.66	2.91	0.71	1.23	0.75	119	13	0.63	2	0.55	0.53	12	5	4	15
98	1.95	2.46	0.67	1.33	0.46	52	12	0.32	2	0.44	0.32	14	5	3	9
99	1.28	2.84	0.64	1.19	0.83	76	11	0.44	1	0.37	0.45	13	6	4	14
100	1.64	1.57	0.7	1.22	0.4	113	12	0.6	2	0.36	0.34	12	5	4	7
101	1.11	1.92	0.64	1.36	0.7	103	13	0.74	1	0.33	0.41	14	6	2	12
102	1.53	2.72	0.75	1.34	0.89	104	11	0.5	2	0.58	0.35	11	5	3	8
103	1.08	2.65	0.63	1.31	0.54	115	10	0.35	3	0.48	0.37	9	7	4	15
104	1.81	2.2	0.74	1.25	0.77	43	12	0.72	2	0.5	0.33	8	4	5	10
105	1.23	1.62	0.67	1.18	0.35	61	15	0.68	2	0.6	0.41	10	7	3	13
106	1.97	2.98	0.55	1.16	0.57	91	14	0.4	2	0.35	0.48	13	6	3	6
107	1.31	2.27	0.56	1.4	0.8	40	13	0.46	3	0.4	0.59	12	4	5	11
108	1.98	1.55	0.62	1.28	0.69	98	11	0.79	3	0.38	0.54	13	7	4	8
109	1.48	1.97	0.52	1.3	0.96	92	10	0.64	2	0.44	0.53	12	5	2	13
110	1.78	2.51	0.59	1.17	0.66	110	13	0.32	1	0.56	0.59	8	6	3	8
111	1.45	2.37	0.53	1.24	0.32	124	15	0.54	2	0.53	0.51	11	4	4	11
112	1.67	1.83	0.51	1.14	0.85	83	9	0.73	1	0.6	0.47	8	6	3	7
113	1.42	2.18	0.57	1.32	0.33	32	12	0.57	1	0.49	0.57	10	3	2	11
114	1.56	2.41	0.72	1	0.3	53	9	0.51	2	0.51	0.52	12	4	3	12
115	1.14	2.6	0.66	1.05	0.68	49	10	0.62	2	0.57	0.6	15	7	4	9
116	1.61	2.16	0.68	0.93	0.38	125	13	0.53	1	0.45	0.58	14	4	5	10
117	1.27	1.71	0.73	1.07	0.88	86	14	0.39	1	0.59	0.47	14	6	4	9
118	1.8	2.39	0.73	0.81	0.74	127	11	0.65	3	0.47	0.55	11	4	2	11
119	1.17	3	0.71	1.09	0.51	82	10	0.71	2	0.32	0.51	10	6	4	6
120	1.59	1.8	0.69	0.82	0.91	59	14	0.52	2	0.43	0.52	11	5	4	13
121	1	1.76	0.66	0.99	0.63	73	14	0.3	3	0.39	0.54	9	6	2	7
122	1.75	2.86	0.6	0.91	0.49	38	14	0.75	3	0.52	0.4	14	7	2	12
123	1.38	2.77	0.5	0.94	0.87	67	13	0.61	2	0.59	0.34	13	5	5	7
124	1.91	1.88	0.55	1.08	0.37	97	9	0.42	3	0.53	0.42	13	7	4	14
125	1.16	2.06	0.54	0.85	0.78	116	11	0.43	2	0.49	0.3	15	4	3	10
126	1.7	2.81	0.6	1.02	0.86	89	15	0.69	1	0.34	0.44	9	6	3	14
127	1.06	2.48	0.57	0.87	0.31	95	12	0.77	1	0.42	0.4	9	5	4	9
128	1.88	1.95	0.61	0.83	0.72	55	10	0.34	1	0.36	0.44	10	7	5	13
129	1.34	1.59	0.54	0.97	0.55	41	11	0.47	2	0.35	0.37	11	5	3	6

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Azimetli, M. *Simulation Of Flight Operations And Pilot Duties In Lantirn Fighter Squadrons Using Simkit.* M.S. Thesis, MOVES Institute, Naval Postgraduate School, Monterey, California, June 2008.

Buss, A. (2002). Component Based Simulation Modeling With Simkit. *Proceedings of the 2002 Winter Simulation Conference,* 243–249.

Buss, A. (2007). Composability and Component-Based Discrete Event Simulation. *Proceeding of the 2007 Winter Simulation Conference,* 694–702.

Buss, A. (1996). Modeling with Event Graphs. *Proceedings of the 1996 Winter Simulation Conference,* 153–160.

Buss, A. (2009). *Mv4302 Advanced Discrete Event Simulation Modeling Class Notes*, Naval Postgraduate School, Monterey, CA.

Cioppa, T. M., & Lucas, T. W. (2007). Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes. *Technometrics* 49(1): 45–55.

Law, M. A. (2007). *Simulation Modeling and Analysis.* New York: McGraw-Hill Book Company.

McLean, C., & Leong, S. (2001). The Expanding Role of Simulation in Future Manufacturing. *Proceedings of the 2001 Winter Simulation Conference,* 1478–1486.

Sanchez, S. M. (2008). Better Than a Petaflop: The Power of Efficient Experimental Design. *Proceedings of the 2008 Winter Simulation Conference*, 73–84.

Sanchez, S. M. (2005). *NOLH Design Spreadsheet.* Retrieved 27 July 2009, from http://harvest.nps.edu

Sanchez, S. M. (2009). Design Of Experiments. *OA4333 Class Notes*, Naval Postgraduate School, Monterey, CA.

SAS Institute Inc. (2007). *JMP 7.0*. Retrieved 26 July 2009, from http://www.jmp.com

Schriber, T. J., & Brunner, D. T. (2005). Inside Discrete-Event Simulation Software; How It Works and Why It Matters. *Proceedings of the 2005 Winter Simulation Conference*, 167–177.

Schruben, L. W., & Yücesan, E. (1988) Simulation Graphs. *Proceedings of the 1988 Winter Simulation Conference,* 504–508.

Schruben, L.W. (1995). Graphical Simulation Modeling and Analysis Using Sigma for Windows, Body and Fraser Publishing Company, Danvers, MA.

Schruben, L. W. (1983). Simulation modeling with event graphs. *Communications of the ACM*, 26(11): 957–963.

The Java Tutorials. Retrieved 17 July 2009, from http://java.sun.com/docs/books/tutorial/java/javaOO/enum.html

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Hava Kuvvetleri Komutanligi
   Bakanliklar-06100
   Ankara, Turkey

4. Hava Harp Okulu Kutuphanesi
   Yesilyurt
   Istanbul, Turkey

5. Kara Harp Okulu Kutuphanesi
   Bakanliklar-06100
   Ankara, Turkey

6. Deniz Harp Okulu Kutuphanesi
   Tuzla
   Istanbul, Turkey

7. Arnold H. Buss, Ph.D.
   Moves Institute
   Naval Post Graduate School
   Monterey, California

8. Susan M. Sanchez, Ph.D.
   Department Of Operation Research
   Naval Post Graduate School
   Monterey, California

9. Serhat Camur
   Hava Kuvvetleri Komutanligi
   Bakanliklar - 06100
   Ankara, Turkey